ROBOTMAKER
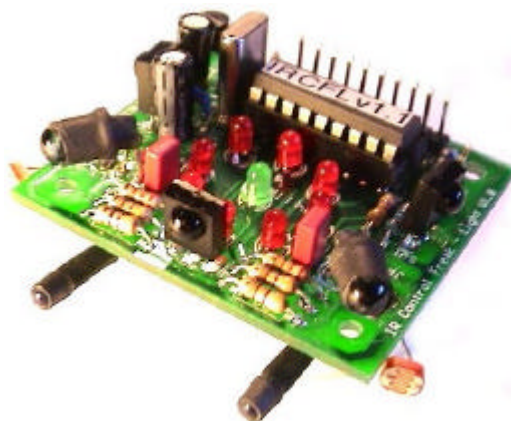
(www. robotmaker.eu.com)

USER MANUAL

_____

INFRARED CONTROL FREAK - LIGHT ™
V2.0

Download Latest Assembly and User Manual from
http://www.robotmaker.co.uk/documentation.htm

ROBOTmaker Ltd
German Lab - Langgasse 71
50189 Elsdorf, Germany
Fax +49 (0) 2274 706554
support@robotmaker.co.uk
www.robotmaker.eu.com

TABLE OF CONTENTS:

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

1   INTRODUCTION

Thank you for purchasing the **Infrared Control Freak - Light V2.0** (IRCF-L) for your autonomous robotics project.

This user manual is relevant for our complete product range of **Infrared Control Freak v.1.1** and the **Infrared Control Freak** - **Light v2.0** product range. Whether kit or fully assembled:

   ?? Infrared Control Freak Light V.1.1 (with two photo-resistor light sensors)
   ?? Infrared Control Freak - Light - Power^2 - (Extended range)

If you bought the kit version, please follow the assembly instructions first.

> **Before starting please read the safety and disclaimer section in Appendix A**

1.1   Upgrade from IRCF v1.0

Based on user feedback, we have made some modifications and improvements to the former IRCF 1.0. All the modifications have been from built from scratch into the new IRCF-L version; including a light sensing functionality.

If you already own the former Infrared Control Freak V.1.0 (i.e. the non-light sensor version), you can benefit by upgrading the firmware to the new V2.0 version. Apart from the new commands dedicated to control the new light-sensors, all the newly upgrade commands will work on the previous IRCF V1.0 PCB versions.

All you need to do is carefully remove the IRCF v1.0 microprocessor, stick it in a piece of polystyrene and return it to ROBOTmaker. We will charge only for the return postage & package. If you are a registered user and have a PIC programmer, you could do this immediately, by downloading the latest firmware version directly from our website.

If you would like to upgrade to the IRCF-L PCB version (with light sensors), we can provide this fully assembled at a special upgrade rates.

1.2   New to Autonomous Robots?

If you are new to autonomous robotics then have a quick look at Appendix B, where there are some hints on how to get started in this exciting new hobby.

1.3   What does the IRCF-L actually do?

Infrared Control Freak – Light™ (IRCF-L) is a multi-functional programmable sensor (IR +Light) and combined infrared Module. The IRCF-L module will connect to many types of robot controllers, such as the Basic Stamp from Parallax Inc, or MEGAbitty from Bittybot.

You don't stop playing because you get old;...
...you get old because you stop playing!

IR-Control Freak LIGHT™ v2.0

Basic Stamp Robot controller

Power Supply and Serial interface cable to IRCF-L

GrowBot Printed Circuit Board
Fitted with Speaker, Batttery, R/C Servo and Power Supply connections

Infrared Control Freak - LIGHT

GROWBOT Chassis

Modified R/C Servo Motor

Infrared Control Freak – LIGHT PCB

IR Emitter

IR Sensor

IR Emitter

Light Sensor

€1 coin

MEGAbitty controller inside tank body

MEGAbitty controller

Infrared Control Freak Light with MEGAbitty robot controller and Toy tank chassis

The Infrared Control Freak – LIGHT programmable module gives you a totally new dimension to your robotics project. All the following functionality is built into one small programmable module:

Infrared Proximity Detection

?? Three Infrared proximity sensors enable 180-degrees of proximity detection. The module acts similar to a range finder, providing the approximate directional and distance of an object, from about 2cm (1") to about 30cms (1ft). 180-degrees proximity detection is useful for robot control through narrow passages; acting similar to an electronic whisker. The module senses the direction of obstacles in a similar way a bat finds its way around in the dark. The main difference is that the

IRCF-L measures infrared light pulses rather than ultrasonic sound pulses. The accuracy depends on the colour of the obstacle (i.e. white reflect almost 100% of IR light while black material absorbs most of the IR light).

?? A compass style LED display 'points' in the direction of the nearest obstacle. A green centre LED illuminates when an object is about 100mm (4") away from the robot.

?? Pre-programmed Fuzzy Logic proximity detection algorithms enable instant object avoidance decision making; freeing up your main robot processor.

?? A programmable option also allows you to build your own Fuzzy Logic rule structure and rule matrix (error/error.dot) from 6 bytes of proximity detection data.

?? The directional and distance data is transmitted via a 3 wire serial connection in ASCII or Decimal format, for easy robot controller interfacing or for merely plotting data on any ASCII terminal screen.

Natural Light Sensors

?? Two Cadmium Sulphide (CdS) photocells are added to the IR-Control Freak - LIGHT version for additional sensing functionality. Your robot will therefore now be able to detect the direction of the brightest light source, enabling your robot to head towards light or 'hide' in the shadows.

?? Your robot can follow a light sources (such as a torch) or you can get your robots to race through a maze with a light source as a direction beacon.

Infrared Remote Control – Infrared Receiver

?? Using a standard Sony TV / Video remote controller you can take full control of your robot and make it do what ever you like. IR-Control Freak™ decodes Infrared commands and converts them into serial data for easy remote control of your robots.

Infrared Remote Control - Infrared Transmitter

?? The module transmits any Sony S.I.R.C Infrared device and button code.

?? Transmitting and receiving encoded SIRC Sony IR signals enables inter-robot communication and identification. This feature can be used for robot gaming e.g. robot tag, strategic war games or for simulation such as Biomimetic approaches & insect navigation

Infrared Beacon

?? A simple IR beacon functionality is implemented to assist navigation. The controller identifies the direction of IR coded signals within 180 degrees. This is used to identify the direction other robots or for locating an IR beacon transmitting a unique code.

?? The module can also be used to detect location of infrared beacons.

Robot Competition

?? The module is compacted onto a PCB that is under 5cm Square. This is therefore suitable for robot competitions as small as Micro-Sumo competitions.

Easy Interface

?? Connect the module to your robot controller using a simple 3 pin serial interface (GND, TX, RX), providing a simple interface to your Basic Stamp or other robot

microcontroller. In-circuit reprogramming is also supported for easy loading new patches and new versions of software (PIC programmer required).

The required functionality is selectable by sending just a single 'byte' command to IRCF-L. If you don't understand any of this, …don't worry as it is all explained in detail in this manual.

Using the examples in this manual (and on the www.robotmaker.eu.com website) you should be up and running, literally within a few hours, without any needing any previous knowledge of electronics or robot programming.

1.4 Pre-programmed proximity commands

The **IRCF-L** is self contained with five infrared proximity sensing command, one light sensing command, 5 infrared communication commands and one mixed proximity and light sensing command; all built into one small module. This 'plug & play' functionality is specially designed for use within multi-autonomous robotic projects such as robot gaming, robot competitions, artificial life simulations, robot swarms and flocking simulation and other research and simulation applications.

1.5 IRCF–L interfaces easily to any microcontroller

IRCF-L can easily be interfaced to almost any robot Microcontroller – Examples are the Basic Stamp from Parallax inc, MEGAbitty, PIC from Microchip, OOPic, PICAXE, Tiny Pod, Basic X24, Handy Board/Cricket from Gleason Research or other robot Microcontroller, etc. The interface uses just two I/O pins and a common ground connection. IRCF-L module is equipped with its own 5v power supply, enabling it to run directly from a 9v PP3 type battery. For more background on the topic "*what is an Microcontroller?*" have a look at appendix B and the website http://imsinet.casa.siu.edu/wam/

1.6 User Support

It is important for us that you have many hours of enjoyment using the IRCF-L module. If you have some questions then just add your comments and details of your project on the forum www.robotmaker.co.uk/phpbb/ or send a question to support@robotmaker.co.uk We will try to respond within 48 hours of your question. Please also add any details of your project, as your contributions are greatly appreciated by other robot enthusiasts. Please also regularly check our website www.robotmaker.co.uk for the latest versions of this user manual and for latest upgrades.

1.7 Tools & Equipment you may need

You will need to make some cables to connect the power supply and a serial connection to your robot controller. It is therefore recommend to have the following tools and equipment:
- Voltammeter or simple circuit tester
- Small Soldering Iron and solder, wire and suitable .1" connectors (for making a cable)
- Wire Cutters
- Safety Glasses
- Ribbon Cable

1.8 Before you start – THINK "SAFETY FIRST"!

Please read the safety instructions and disclaimer in **Appendix A**, before starting to use the IR Control Freak - LIGHT.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

2 Overview of IRCF-L functionality

2.1 Microprocessor controlled sensor and IR module

The brain of Infrared Control Freak is a PIC16F628 microprocessor from Microchip. PIC stands for 'Programmable Intelligent Controller'. This is an 8-bit micro controller with a Harvard architecture; using separate buses for the data and instructions. It has built in integrated program memory, volatile and non-volatile RAM, 13 IO ports, timers, IO-pins to perform analogue actions, such as comparators, AD-converters and DA-converters. With a 4MHz crystal, this device executes 1 million instructions per second.

2.2 Microprocessor Firmware

'Firmware' can be compared to the 'basic instincts' and subconscious routines that are pre-programmed in the brain at 'birth'.

ROBOTmaker have pre-loaded a programme into the IRCF-L's microcontroller 'brain' which manages all the communications to your robot microprocessor and executes all the selected command routines. This program is called 'firmware' as it is permanently written to the chips internal flash memory. It will not disappear when the power supply is removed. The microprocessor's memory can only be deleted or overwritten by using a special PIC programmer.

If you are a registered user and have a PIC programmer, then it is possible to use the provided 'In-Circuit Programming' connection to upload new versions of the firmware or any specially developed variants of the firmware. Alternatively, ROBOTmaker can load new versions of the firmware for you by returning your microprocessor.

2.3 Non-contact proximity detection and distance measurements

The **IRCF-L** offers non-contact distance measurements from about 2cm (1") to about 30cms (1ft), by sending short bursts of Infrared pulses from the left and right Infrared LEDs.

The number of reflected 'hits' detected by each IR sensors is decoded and the approximate direction and distance of an obstacle is calculated.

For most Infrared Proximity Detection (IRPD) applications, an optimal range of 30cms has been selected to suit most 'desktop' style robot projects. Only objects this range and heading towards the IRCF-L are of immediate interest to the robot. Object further away, or passing-by, can normally be ignored.

For larger size robot applications, where a longer-range is required, see the Infrared Control Freak - Light – Power² product range, which has an adjustable range control.

2.4 Infrared Proximity Sensing Envelope

The three Infrared proximity sensors to provided to give approximately 180 degrees of detection. The combined directional and distance sensing data provides excellent autonomous control through narrow passages or mazes. Using the "proximity detection" command code 32 (see command code overview section below), the IRCF-L returns with 6 bytes of data via the serial port, containing information about the obstacle's approximate directional and approximate distance data.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

IR-Control Freak LIGHT™ v2.0

VERSION ICRF-L v2.0
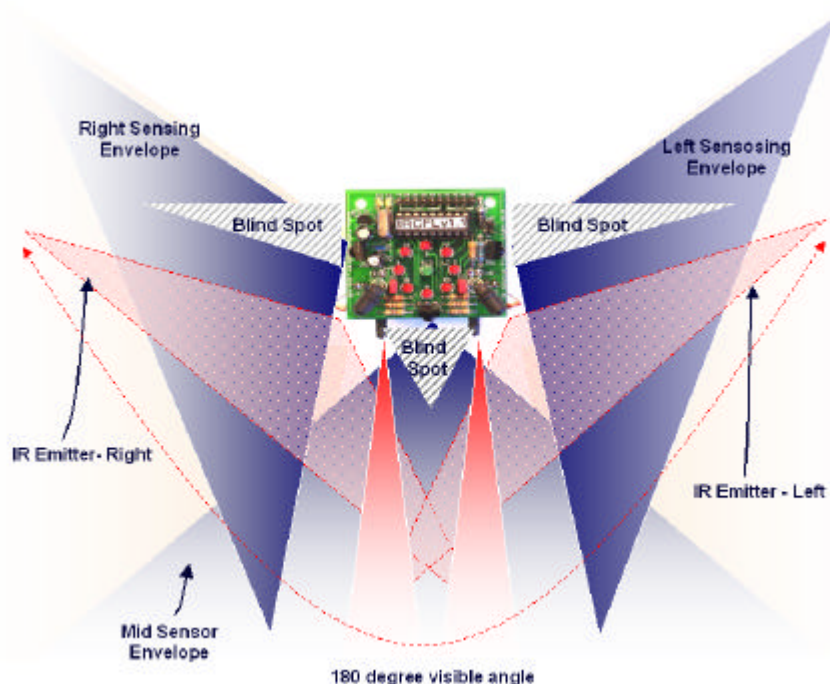
This is dependant on how the sensors and Infrared LEDs have been calibrated. Details of how to calibrate the module are detailed in Appendix C – Module Calibration.
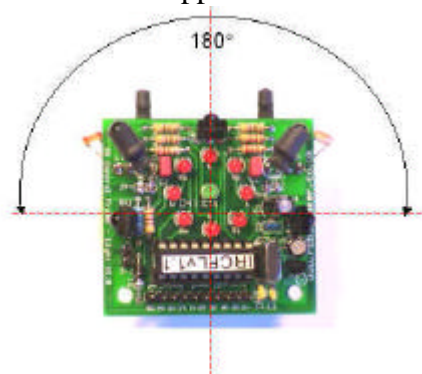
2.5   Infrared Proximity detection - Directional Indication

The circular 'compass' style LED-display, 'points' in the direction of the nearest obstacle (within a 180 degree arc) during the 'Infrared Proximity Detection' (IRPD) routine.
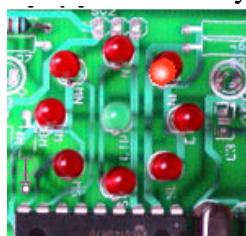
The green centre LED will illuminates when an object is about 100mm (4") from the device.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

Centre **LED** indicates close proximity of object

Red **LED** Indicates direction of nearest object

Directional information of localised obstacles offers instant decision making during obstacle avoidance, maze running and robot gaming. For example, objects that are not directly ahead are ignored or using your own fuzzy logic routines, you can track an object and determine whether it is getting closer or further away.

2.6 Visual Feedback, based on Mode of operations

The circular display is also useful for providing visual feedback on the state of the program being executed. Each command will display a different display pattern so you can quickly debug your program by seeing the current routine being executed. Details of the display patterns for each command are provided in the command overview section.

2.7 Fuzzy Logic

The pre-programmed Fuzzy Logic proximity detection algorithms enable instant object avoidance decision. This frees-up a lot of processing power that your main robot controller would normally need to do.

If you don't want to use the pre-programmed logic, you can also experiment and design your own Fuzzy Logic algorithms, based on the six bytes of 'raw' proximity detection data.

By interpreting the data from each of the sensors over a period of time, it is possible to determine whether an object is moving towards / away from the robot. Further details of how to create your own Fuzzy Logic routines are detailed on **Appendix C – Overview of Fuzzy Logic.**

2.8 Infrared Remote Control

For remote control of robots and electronic projects using a standard TV remote control, the **IRCF-L** is equipped with an onboard Infrared remote controlled receiver and transmitter.

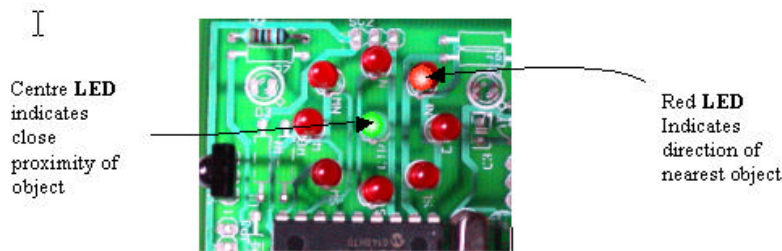The protocol used with **IRCF-L** is the Sony format known as Sony SIRC Protocol. See **Appendix B – Overview of Principal of Infrared Remote Control** for more details of the various device and button commands available to be used.

2.9 Inter-Robot communications

By using simple coded IR messages, it is possible to transmit signals to other robots. For example, it is possible to determine whether an opponent robot is a friend or foe. You could also configure a unique ID for each robot and robot team, allowing your robot to communicate between team members and sending a 'tag' or 'zap' code to the other opponent team. For example, ten 'zaps' of device code/button code 10,10 to the centre sensor could mean, 'your it!' (e.g. in a game of tag) or "your dead!" (e.g. in robot battle game)

2.10 Customise an Infrared Control Freak - LIGHT

If you require special functionality, special protocols or other special control characters for your robot project, just send an email to [support@robotmaker.co.uk]**,** or add a comment to the forum: www.robotmaker.co.uk/phpbb/ for discussion. We will try to add this for you too or create a customer version.

2.11 Modifications for increased sensing range.

If you own the IRCF 1.0 or IRCF-L 1.1, the range can be increased by modifying the IRCF-L to achieve an extend range of about **1 meter** (1yard). This modification is only recommended for students with a basic electronics understanding and some experience with soldering. See APPENDIX J - Modifications for increased range.

The IRCF-L has been designed to provide optimum sensor distance for most robot projects. However, in a very few circumstances, an increased sensing range may be required for certain experiments. If you only intend to run longer-range experiments, then we recommend that you purchase the Infrared Control Freak POWER[?] range of infrared sensors that are designed for longer range sensing. (Production versions will be available from end 2004).

2.12 Calibration of the Infrared Control Freak

As the infrared emitters and sensors can easily be moved out of position, it is recommended to calibrate the IRCF-L before initial use and before each exercise. See appendix C on details of how best to calibrate the module.

ROBOTMAKER
You don't stop playing because you get old;...
...you get old because you stop playing!

3   Serial interface cable connection

A two-pin I/O serial connection (including a common ground) is required to connect the IR Control Freak module to almost any microcontroller.

Pin Header

Pin 6
(Serial TX)

Pin 5
(Serial RX)

Pin 4
(GND2)

Pin 3
No
Connection

Pin 2
(GND 1)

Pin 1
(9v)
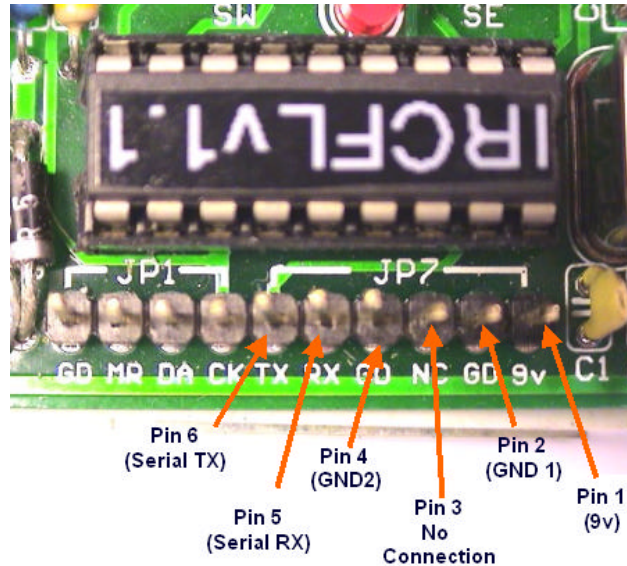
Connector pins 1-2 are for the 9v power supply connection. It is important to **pay attention** to the polarity of the power supply; otherwise the IRCF-L microcontroller will be destroyed! To prevent connecting the connector the wrong way, just place a peg in Pin 3.

Pin inside
Female header
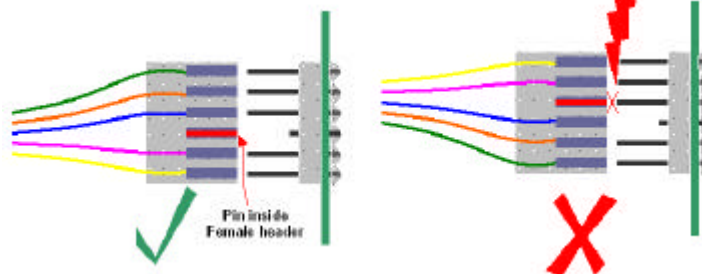
Pins 3-5 are for the serial interface connection.

Pins 7-10 are provided for in-circuit programming and are only required to load new versions of the firmware into the IRCF-L Microcontroller. A special PIC programmer is required for this. If you don't have a programmer, just remove the chip from the socket (take care not to bend any pins when taking this out of the socket) and return it to ROBOTmaker. We will re-load it with the latest software version. More details, on how to download the latest Firmware and references to 'in-circuit programming' are covered on the ROBOTmaker website.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

Pin No.	IC Socket Pin	IC PIN Name	Description
1	NA Regulated 5v supply to pin 5	NA Regulated 5v supply VDD	9V SUPPLY (or regulated 5v supply from other source).
2	5	VSS	GROUND – for power supply
3	NA	NA	Not connected. This pin is used to prevent incorrect connection of connector (see diagram above)
4	5	VSS	GROUND – for serial interface
5	18	RA1	Serial connection RX - Pin A1 on PIC. This is connected to the data transmit (TX) on the main robot controller
6	17	RA0	Serial connection TX – Pin A0 on PIC This is connected to the data receive (RX) on the main robot controller

3.1   Serial cable connection To PC

Disclaimer: It is important that you first check that all the cables are connected the correct way around **BEFORE** connecting the 9v power supply or connecting to your PC.

It is particularly important you check that the voltages and currents from the IRCF-L will not destroy your robot controller or your PC! Check also that the voltages from your PC's serial connector will not destroy the IRCF-L. Connect at your own risk!

To construct the interface cable, you will need:
-   Female pin header block (6 way)
-   9 way D-Type Plug (Female)
-   Ribbon Cable or wire
-   1Kilo ohm resistors x 2 (voltage limiting)
-   9v PP3 style Battery terminal
-   Hot Glue-Gun (not necessary but recommend)

Connect the cable from the IRCF-L to a serial connector as follows:

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

IR-Control Freak LIGHT™ v2.0
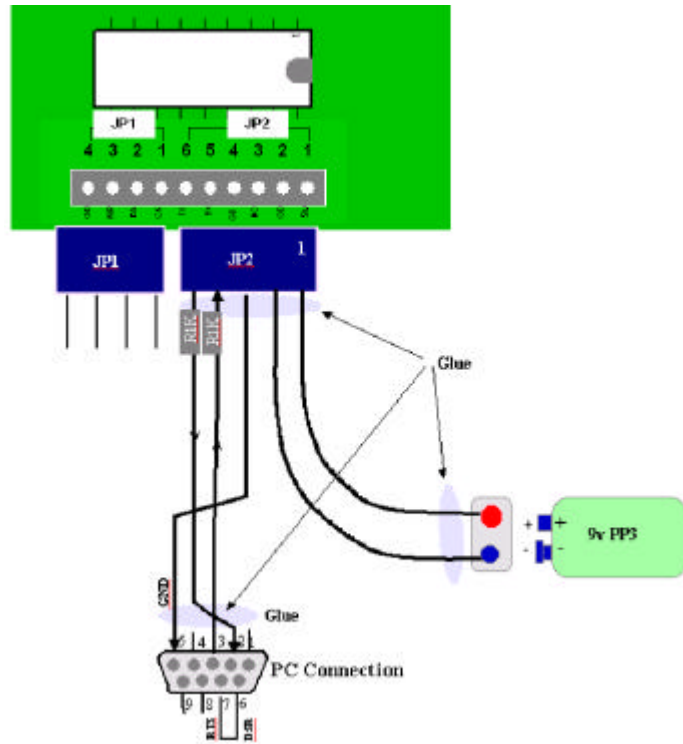
The 2 x 1K resistors are voltage-limiting resistors. These may be required depending on the configuration of your PC's communication port. Double check that the connector pins are connected correctly (see table above). Please check this before connecting.

Finishing off

If you intend to do a lot of testing, then it is recommend putting heatshrink around the cables and connector, or using a glue gun to stick the cables together. This will help prevent the cables breaking.

3.2   Serial connection to the main robot controller

Using auxiliary power supply (from 9v battery)

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!
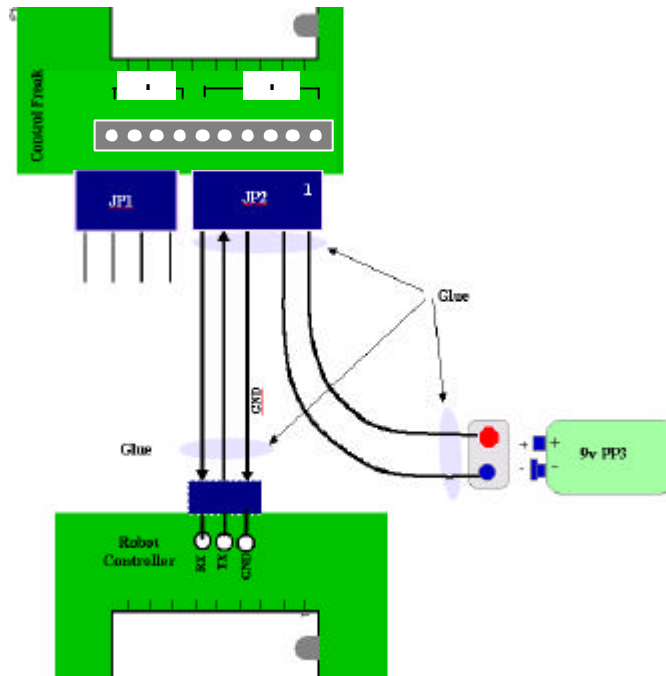
3.3   Using a 5v regulated power supply from the Robot Controller.

IRCF-L consumes on average 10mA-15mA in normal use (depending on how many LEDs are illuminated). Before connecting the **IRCF-L** to your robot controller's power supply, it is important, that the current consumption is tested and confirmed and that your robot controller can handle this additional loading. Please check the reference documentation provided with your Microcontroller before connecting. You connect at your own risk!!

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!
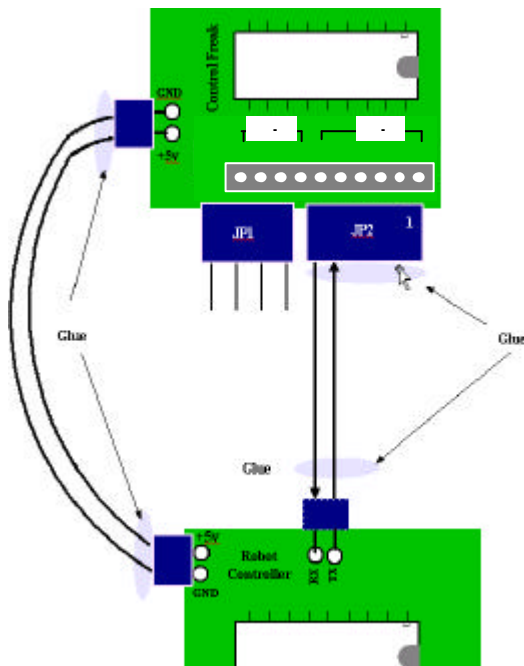
4   Testing the connection to the PC using communication software.

There are many freeware, shareware and commercial software programs to test the serial connection with the Infrared Control Freak Light and with the robot controller.

One program that can be recommended is the software from Docklight. See (http://www.docklight.de/). Evaluation copies of the software (with limited functionality) can be downloaded from their website for free. The demo version works fine for limited testing of the Infrared Control Freak. You will need to set-up the communication speed, by going into the Tools/Project setting option:
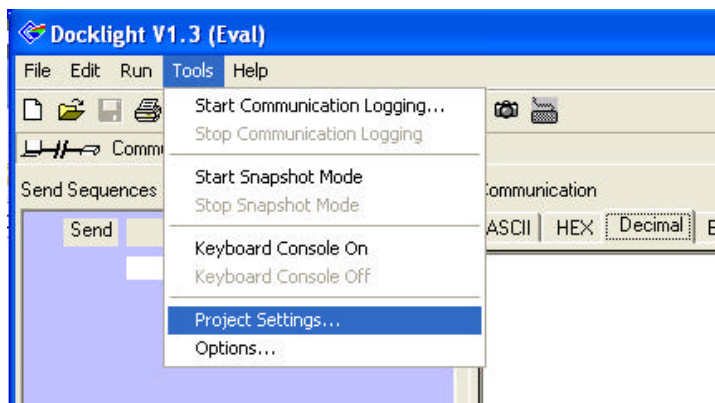
Then set the communication speed as follows:
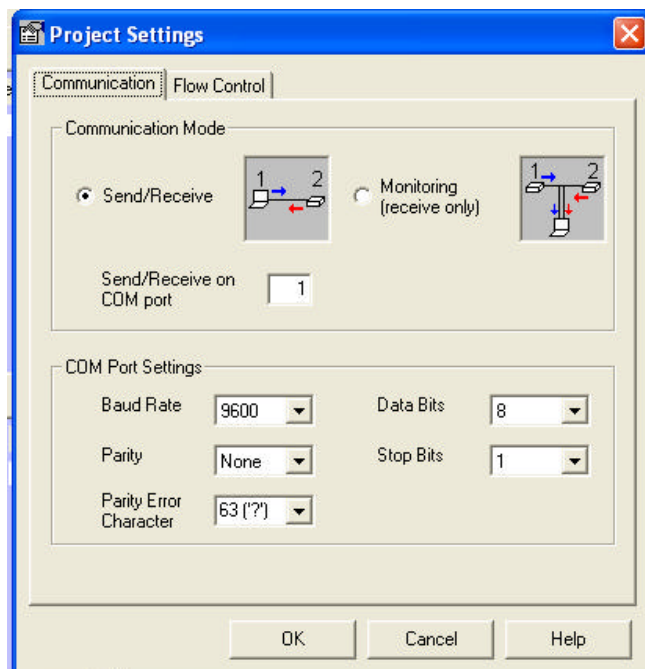
On the flow control tab, set flow control to **OFF**

If you decide to download this software, create a blank project and start by creating a 'send' command button with the value '034' or '044' (Freewheeler command) as follows:

Create a 'Send Sequence' by clicking the name row.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!
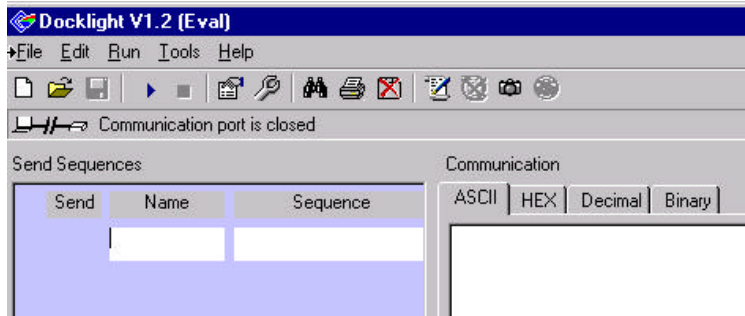
A pop-up window will appear to enable the send sequence button to be created.
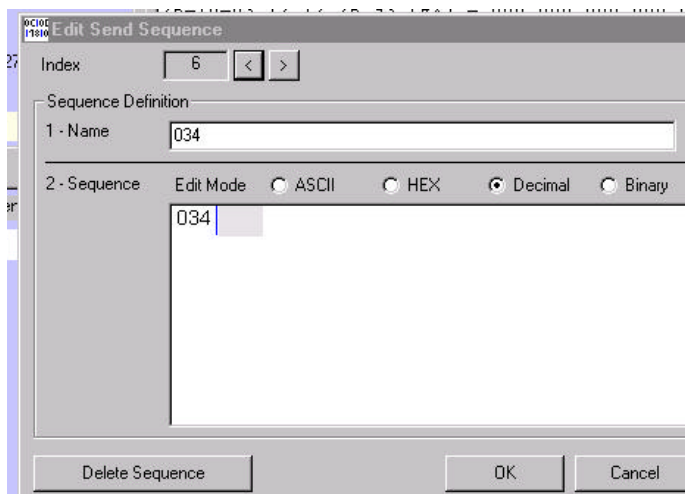
Enter 034 as the sequence name and sequence command and press OK. A 'Send Sequence' button will be created. Press this button and the value DEC 034 is sent to the **IRCF-L** .

A continuous stream of proximity output data will be transmitted to the PC screen as follows:
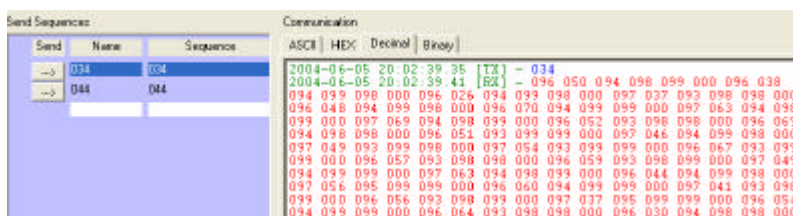
The circular LED display will light up in the direction of the nearest obstacle. To exit from this mode, send a different command code (e.g. 032).

Other 'send sequence' buttons can be made in the similar way. Try 044 and you get the same in ASCII format (explained more below). This is a good way to get familiar with the commands.
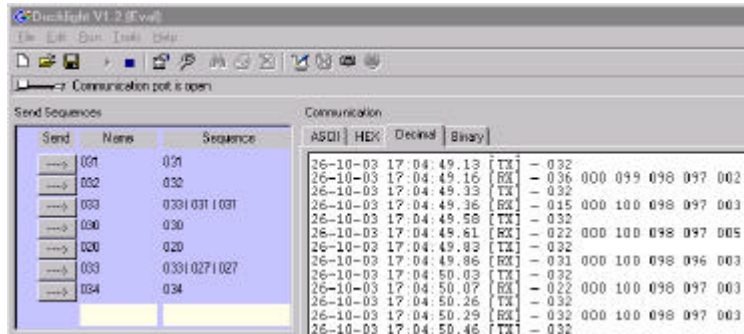
HyperTerminal

HyperTerminal is another software application that can be used. It is most probably already installed on your PC (see under Programmes/Accessories/Communications). The HyperTerminal also works very well when sending command code to the Infrared Control Freak. To learn more about ASCII format see appendix E– ASCII TABLES or htpp://www.asciitables.com. The main reason some commands have an ASCII output option is that it is a lot easier to interface to an ASCII terminal emulator and debug the results. These commands are also more likely to be used in standalone non-robot applications.

Press 'Connect to' and select COM1 (or what ever COM port you have connected the **IRCF-L** to). Press OK.

In the COM1 property window, select 9600 baud, 8 data bits, no parity, one stop bit, and flow control as 'None'

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

IR-Control Freak LIGHT™ v2.0
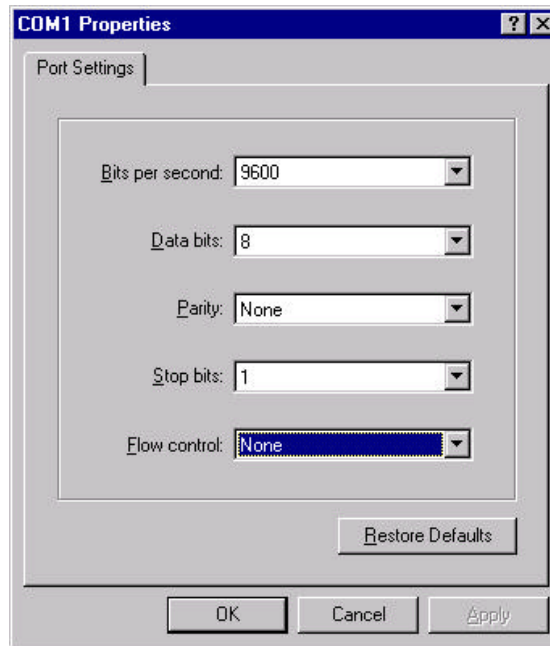
The command 'sent' to the **IRCF-L** needs to be in DECIMAL format. There are no normal character keys that can be pressed on the keyboard that represent 031, so this must be sent by holding down the ALT key and typing 031 on the keypad (this must be done on the keypad, with the num‑lock on). Then release the ALT key.
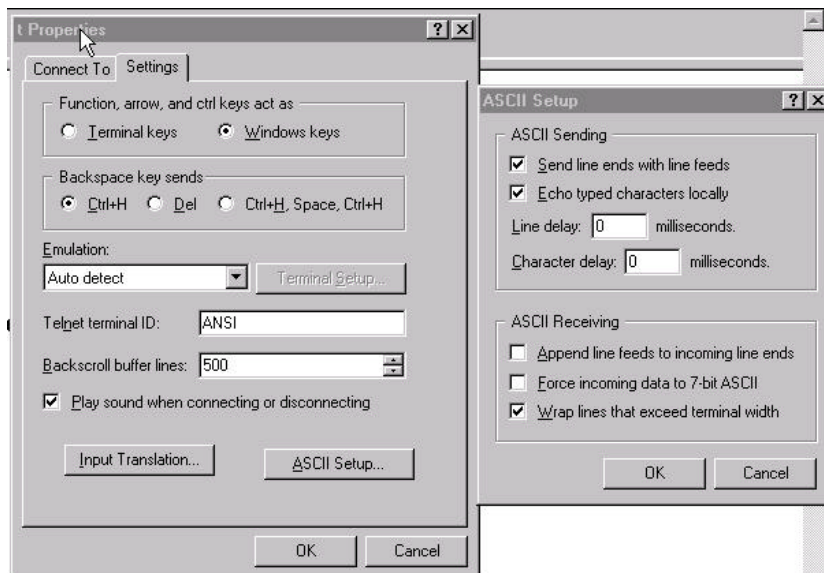
If nothing happens, you can set the 'ASCII Sending' property as defined above and see what is being sent. A graphic symbol should be displayed something like this ? .

Press a button on the remote control and the device code and button code will be displayed on the main HyperTerminal window.

- 18 -

5   Power-up tests

The **IRCF-L** module is programmed with a power-up routine. Connect the module to a 5v regulated supply or a 9v supply as described in section 3.1, 0 & 3.3

Within a few seconds of connecting the power supply, the red LEDs should light up; one after the other, in a clockwise direction and then in an anticlockwise direction. The module will then search for a command on the serial port. If no command is being transmitted, the module will continue this routine until a command signal is detected.

You don't stop playing because you get old;...
...you get old because you stop playing!

6   Programmable command codes

The **IRCF-L** is designed for many types of robot or electronic projects. This flexibility has been achieved by defining pre-programming routines that are activated by a single command code. The following command codes are available and explained in full detail within this next section:

CMD Code	Mode Name	Description
020	Infrared Transmission (ZAP/TAG Mode)	Transmits a Sony (SIRC) Infrared command from both IR LED's. Any Device + Button code can be transmitted.
		Decimal output
030	Sony IR -Decimal	Receive Sony (SIRC) IR Commands and output results to serial port as two-bytes in decimal format
032	IRPD Mode	One shot proximity sensing. 6 bytes of serial data are transmitted with proximity information in decimal format. This is the normal command used for IRPD.
034	Freewheel Mode	Send a continuous stream of proximity serial data in decimal format.
035	Autopilot Mode	Using the internal Fuzzy logic routines, results are sent as two bytes of data to the serial port. These bytes depict the direction and distance of the nearest obstacle.
036	Beacon Mode	This command searches for Sony (SIRC) infrared transmissions on all 3 sensors and returns with an approximate direction based on which sensor received the highest readings.
037	Light Sensor	Checks the left and right light levels and outputs the results as two bytes of data in decimal format. The circular display also indicates the approx. direction of the brightest light source.
038	Light Sensor + IRPD	This command is a combination of IRPD Mode (037) and Light Sensor Mode (037). 8 bytes of data are returned. The command enables IRPD data to take priority over Light sensor data.
		ASCII output
040	Sony IR -ASCII	Receive Sony (SIRC) IR Commands and output results to serial port as two-bytes in ASCII format
042	IRPD Mode	Same as command code 032, but returns results in ASCII format.
044	Freewheel Mode ASCII	Same as command code 034, but is sent as ASCII format for direct connection to ASCII terminal
		Other useful Commands
052	Continuous Infrared transmissions (Beacon transmission mode)	This command transmits a REPEATED Sony (SIRC) Infrared transmission command from both Infrared LED's. Similar to command 020, but automatically re-transmits the command every 2 seconds. Any (SIRC) Device code + Button code is transmitted. This command is intended to be used together with 1 or many Beacons sending different ID codes to guide bots towards them. It can also be used for debugging other infrared devices. The IRCF-L can be disconnected from the serial connection after being 'primed' with a device code and a button code. It will then continue to re-transmit the initial command, until another command is sent.

Table 1

6.1 Sending commands to the IRCF-L and receiving results using the basic stamp from Parallax Inc.

It is very easy to send and receive commands to/from the IRCF-L. In the examples below, the Basic Stamp (from Parallax Inc.) has been used to demonstrate how a simple interface can be established. The Basic Stamp II is only an example robot Microcontroller and does not necessarily imply any particular preference. The main advantages of the Basic Stamp for beginners, is that the Basic Stamp II has an in-built Basic Language Interpreter. Basic Language it particularly easy to learn and therefore will get you up-and-running very quickly. Parallax also provide other versions of the Basic Stamp that provide faster speeds and other interpreters, such as Java (BSp). If you are not using a Basic Stamp, you will need to refer to your own Microcontroller's user reference manual for instructions on how to set up a serial interface.

There are some rules to take into account when setting up the communication interface. These rules are to ensure the Infrared Control Freak and your robot Microcontroller can understand the data being transmitted or received.

A) The command sent to the IRCF-L, must be a 'decimal' value.
B) The data RX/TX rate must also be set to 9600 baud (bits per second), 8 data bits, no parity, 1 stop bit. Flow control should be set to OFF.

Decimal or ASCII data can be received from the IRCF-L, depending on the command you chose. Normally, the decimal values are used for connection to a robot controller and ASCII values are used for connection to an LCD or terminal program (e.g. Hyperterm, StampPlot).

Sending commands to the IRCF-L:
An example Basic Stamp* command (*from Parallax Inc):

```
SEROUT 2, 16468, [32]
```

This command means - One byte of data will be sent out on pin number two (assuming pin two is connected to the input pin ["RX" pin] of the IRCF-L). The code '16468' is just an internal instruction within the Basic Stamp, setting up data transmission at 9600 Baud, 8 bits, no parity, inverted (i.e. direct connection). The value in parenthesis is the data to be transmitted to the Infrared Control Freak. In this case, the value '32' is sent.

Reading Data from the IRCF-L :
To read data from the Infrared control freak is just as easy. The example below reads two bytes of data from the IRCF-L to the Microcontroller:

```
SERIN 4, 16468,1000,TIME_OUT, [DEVICE_CODE, BUTTON_CODE)]
```

This Basic Stamp command means – Receive bytes of data on pin number four. The code 16468 is just an internal instruction within the Basic Stamp to read the data at a speed of 9600 Baud, 8 bits, no parity, inverted (i.e. direct connection). The program will jump to a subroutine called TIME_OUT (shown in example below) if data is not received within 1000ms. The data values received are assigned to the variables within the parenthesis (in this case there are two variables as bytes called "device_code" and "button_code"). These

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

variables need to be assigned at the beginning of your Basic Stamp program as type 'byte' or 'word'.

These principles for sending commands to the **IRCF-L** are the same for almost any robot controller.

Below is an example of a complete Basic Stamp program using the Infrared Proximity Detection (IRPD) command 032. In this example, byte 032 is sent to the IFCF-L to trigger the IRPD command. The IRCF-L will respond with 6 bytes of proximity data. This can either be read as 6 bytes in one long string (i.e. extending the example above) or by reading one byte at a time and placing the results into an array. The example below also shows how easy it is to set-up an array, using the Basic Programming language contained within the Basic Stamp* (from Parallax Inc.).

In the following sections each command, code is explained in detail. Examples programs and applications are discussed in the forum at http://www.robotmaker.co.uk/phpforum/.

Examples and videos can also be downloaded from the website:
http://www.robotmaker.co.uk/IRFC/examples

Example:

```
'{$STAMP BS2}
'Test Command 032 - Single Shot

RX              VAR BYTE(5) 'Array with 6 bytes of data from 0-5
DATA_ARRAY      VAR BYTE

'**************************
'* MAIN PROGRAM *
'**************************
MAIN: 'A label called MAIN
GOSUB COMMAND_032
GOTO MAIN


'**************************
'* SUBROUTINE *
'**************************
COMMAND_032:    'A subroutine label called COMMAND_032

'Send the command 032 to the IRCF-L
SEROUT 2, 16468, [32]

'Read serial port and receive 6 bytes of proximity data from IRCF-L
FOR DATA_ARRAY=1 to 6
SERIN 4, 16468,1000,TIME_OUT, [RX(DATA_ARRAY)]
NEXT        'Read next value
RETURN       'Go back to main program routine and start again


TIME_OUT: 'A subroutine label called TIME_OUT.
'If no serial data is received then the SERIN command the program will
jump to this subroutine and print a warning on the Basic Stamp debug
screen
DEBUG "TIMED OUT", CR
RETURN         'Go back to main program routine and start again
```

6.2 Command Code 020 – Sony SIRC IR Transmissions Mode. (Used for inter-robot communication, TAG/ZAP gaming and Beacon transmission mode)

Syntax: 020 <button Code><device Code> (decimal format)

Operands: <button Code><device Code>

Response from IR_CF: An 'X' pattern will flash on the circular display for each IR pulse transmitted.

Overview: IRCF-L Transmits a Sony (SIRC) IR message from both IR LED's.
This command is used to send simple commands to other robots or Sony SIRC devices (e.g. television, video, etc) and used in conjunction with command 030 (receive Sony IR commands). This command transmits SIRC Infrared data from the IR Emitters.

The objective of this command code is for two or more robots to communicate with each other in a very simple manner. It can also be used as an Infrared Beacon for robots to navigate towards the strongest IR signal.

For this command to work correctly, your robot controller needs to transmit a 3-byte string.

The first byte is the command mode instruction (020). The other two bytes represent the Sony infrared 'device and button' codes (see example below).

As soon as two bytes are received, IR-Control Freak decodes, and transmits these two bytes as SIRC Infrared 'device and button' code data. It is as though you had pressed a button on an Infrared remote control device.

TAG/ZAP mode
In multi robot gaming with two teams of robots, the infrared data could be used to identify which robot was sending the message and in which team the robot belongs. This is achieved by assigning each robot a unique device code. For example:

Device code	Representations
11	Team 1 – Robot 1
12	Team 1 – Robot 2
13	Team 1 – Robot 3
21	Team 2 – Robot 1
22	Team 2 – Robot 2
23	Team 3 – Robot 3

In a game of robot tag, the robot with device code number 11 could be classified as the 'it' robot, so all other robots can avoid this robot.

ROBOTMAKER
You don't stop playing because you get old;...
...you get old because you stop playing!

The button code could be used to send a unique instruction such as a 'ZAPPING' or 'TAG' command. For example, robots that receive 10 x ZAP codes could be out of the game for a defined period. For example, after 5 minutes the robot could 'recover' and join the game again. For this to work, the same game rules and code, will need to be provided by the club and applied to both robot teams. There are many university project reports on the subject of robot gaming. See our website for more details.

In conjunction with a radio modem the robot controller could send details to a central PC indicating which robot is 'IT' (e.g. in a game of tag) and which robots have been ZAPPED.

Control of Sony (SIRC) Devices
This command code could also be used for controlling a Sony SIRC device (e.g. Video Recorder, Television, DVD, CD, etc.). As a party trick, you could get your robot to navigate to your to Hi-Fi and turn the music down when it gets too loud!

Example Subroutine:

```
IR_TX_MODE:

'Setup button code and device code by placing values into an array
        BUTTON_COCE = 026                          'BUTTON CODE
        DEVICE_CODE = 025                          'DEVICE CODE

'Send the Button Code & Device code as string

        SEROUT  2, 16468, [20, BUTTON_CODE,DEVICE_CODE]

RETURN
```

VERSION ICRF-L v2.0

6.3   Command Code 030 & 040 – Sony Infrared Receiver Mode

Syntax: [030] (decimal format)

Response from IRCF-L: [Button code], [Device Code]
The response will be in decimal format if 030_command code is used or in ASCII format if
the 040_command code is used.

The green LED on the Circular display will illuminate to indicate the Sony IR mode as been
triggered and the **IRCF-L** is searching for Sony IR messages. The module will remain in
this mode until the robot controller sends an alternative command or if the power-on
infrared signal (17,21) is transmitted. If the IR transmission corresponds to one of the
reserved button codes (detailed in the table below), the corresponding red LED will also
illuminate. See table below:

<div align="center">Reserved Button Codes</div>

Purpose of Command	Circular Display	Sony Button, Device code	Remote control Button name
Exit routine	Exit Command Mode	17,21	(Power On\Off)
Turn Left	Left LED (West)	17,58	(Fast Rewind)
Turn Right	Right LED (East)	17,59	(Fast Forward)
Forward	North LED	17,18	(Volume Up)
Reverse	South LED	17,19	(Volume Down)
Stop	Centre LED	17,56	(Stop)
ZAP	Arrow North flashes	17,9	(Channel 0)

*The actual device codes and button codes may vary depending on the manufacturer of the
remote control unit.

<div align="center">Table 2</div>

Overview:
This command receives Sony (SIRC) IR signal. The IRCF-L module decodes the infrared
signals as two-bytes (button code and device code) in decimal format. The IR signal is
detected by the central sensor only, to facilitate more directional control in robot gaming.
See also the Beacon Command mode for additional functionality.

This command can be used for many control projects. One use of this command code is for
where two or more robots need to communicate with each other in a robot tag game. See
TAG/ZAP mode in 020_command example above.

Infrared Remote Control of your robot
This command can also be used to control the robot via a standard multifunction
TV/VCR/DVD/CD remote control unit. These controllers would also normally have the
SIRC protocol (e.g. Sony AUX device). When configured, you can use the remote controller
to manually take over control of your robot.

When the command code 030 is transmitted from the robot controller to the **IRCF-L**
module, the 'Infrared Command Mode' is triggered. This means the module will keep
searching for a SIRC Sony infrared remote control signal from a TV remote control unit or
other any other robot attached with an **IRCF-L** module.

If the module receives a signal, the **IRCF-L** will decode the Sony Infrared button and devices codes received and transmit these to the robot controller. You can use these codes to control the movement of the robot. Any reserved buttons, as specified in Table 2 above will illuminate the corresponding LED on the circular display. For example, suppose that you would use the button codes [Fast Forward] to turn your robot right, the robot controller will be programmed to accept 17,18 as a command to turn right. Likewise, [Fast Rewind] to turn your robot left and.[Volume Up] to travel forward and [Volume Down] to go backwards.
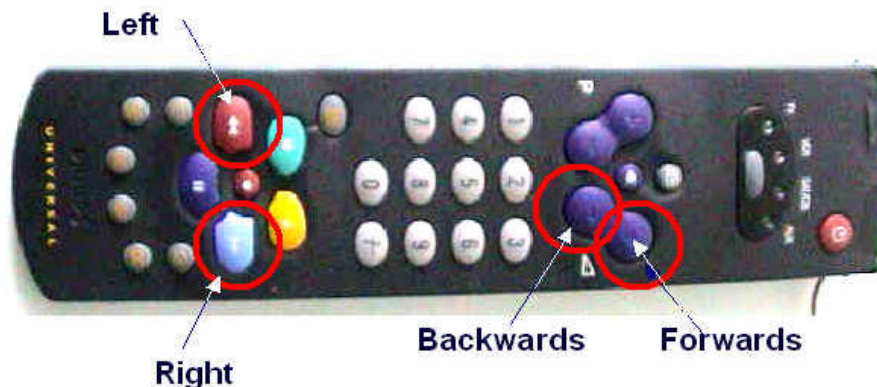
You could of course use any other button to carry out any routine you like, for example you could also just use the number pad for Left. Right, Forward, Reverse.

Resetting the Command
To reset the **IRCF-L** back to the default 'standby' mode, just press the [Power] button on the remote controller (17,21) or send another command (such as DEC 032) from the master controller.

Example Subroutines:
Example code for Basic Stamp (from Parallax)

Sending the command to **IRCF-L:**

```
SONY_IR_COMMAND_CODE:
        ' Send control code 030 at @9600 baud
        ' From [pin 2] (could have been any pin), [9600baud], [dec 32]
        SEROUT 2, 16468, [030]
RETURN
```

The above example is a Basic Stamp program code where the format is:

SEROUT[pin number on Basic Stamp], [baud rate], [decimal character to send]

Receiving decoded data from **IRCF-L:**

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

```
CHECK_IR_CODE:
        'Get 2 Bytes of data
        SERIN  4, 16468, 5000, TIME_OUT, [DEVICE_CODE, BUTTON_CODE]
        'Show results on debug screen
        DEBUG "DEV=", dec DEVICE_CODE, " BUT=",dec BUTTON_CODE
        DEBUG CR
        'get robot to move L, R, F, B depending on what button was pressed
        If BUTTON_CODE = 18 then IR_FORWARD
        If BUTTON_CODE = 19 then IR_BACK
        IF BUTTON_CODE = 59 then IR_RIGHT
        IF BUTTON_CODE = 58 then IR_LEFT
        IF BUTTON_CODE = 5 THEN IR_STOP
GOTO CHECK_IR_CODE ' keep looping forever
```

2-bytes of data are returned containing the button code and device code respectively, in decimal format.

Example using Docklight program
In the example below, a DEC 030 was sent to the **IRCF-L** (TX). This activated the Sony IR receiver mode (in decimal format) and the central green LED on the circular display will illuminate, to indicate that the command is activated. When pressing the fast forward button on the remote control, the **IRCF-L** displays the device code (017)* and the button code (059)*. The right (east) LED will also illuminate on the circular display.

Note that to see the results in the correct format you will need to press the ASCII tab.

You can customise the reserved commands when ordering the **IRCF-L** . Please specify any specific requirements at the time of ordering.

Resetting the IRCF-L command: The **IRCF-L** will remain in the IR receive mode, unless a button code 21 (power on/off) is transmitted or a different command transmitted by the robot controller.

To learn more about ASCII format see appendix E- ASCII TABLES. The benefit using ASCII format is that any ASCII terminal emulator, such as HyperTerminal can be used to display the results.

You don't stop playing because you get old;...
...you get old because you stop playing!

6.4 Command Code 032 & 042– Infrared proximity detection mode (IRPD)

Syntax: 032 (decimal format)

Response from IR_CF: This command forces the **IRCF-L** to execute a 'one-shot' proximity search. **IRCF-L** responds with 6-bytes of serial data, containing proximity information. This is the main infrared proximity command.

The IR control freak can send decimal or ASCII data format depending on the command code transmitted (032 or 044 respectively).

The data stream returned from the IRCF-L, would look something like the following example:

Example:
Command transmitted by the robot controller: SEROUT 2, 16468, [**32**] (Basic stamp code)
Data Returned back from IRCF-L to the robot controller: **50,2,25,1,0,0**

What does this actually mean? - It mean that the right sensor is receiving 50 hits from the right LED and 2 hits from the left IR LED. In addition, the middle sensor is receiving 25 hits from the right LED and 1 from the left IR LED. No IR reflected hits are picked up by the left sensor. This means that the obstacle is about 100mm away, situated on right at about 60 degrees.

Proximity information will be between the range 0-100; where '0' indicates 'no obstacle' detected and 100 indicates that the object is <50mm (2'') away from the sensor.

The 6-Bytes contain proximity information related to sensors that detected proximity. Similar to the concept of stereovision, by decoding data between the sensors it is possible to estimate the approximate position of the sensor.

RIGHT SENSORS		MIDDLE SENSOR		LEFT SENSOR	
RHITS_R	**RHITS_L**	**MHITS_R**	**MHITS_L**	**LHITS_L**	**LHITS_R**
Pulses received by the right IR sensor and transmitted from the RIGHT IR emitter.	Pulses received by the right IR sensor and transmitted from the LEFT IR emitter. The object will need to be very close for readings to be made.	Pulses received by the middle IR sensor and transmitted from the RIGHT emitter.	Pulses received by the middle IR sensor and transmitted from the LEFT IR emitter.	Pulses received by the LEFT IR sensor and transmitted from the LEFT IR emitter.	Pulses received by the middle IR senor and transmitted from the RIGHT IR emitter. The object will need to be very close for readings to be made.

The Circular display will also illuminate LEDs in the direction of the nearest obstacle.
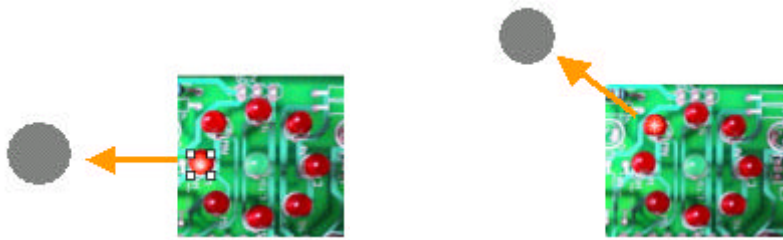
The 6-bytes of proximity data returned to the main robot controller can be divided into three sets of 2-byte pairs; one set for each Infrared sensor. The data sets are related to information reflected back from the RIGHT Infrared LED and the LEFT Infrared LED. This is used to determine the direction and distance of an object as follows. In certain situations, infrared pulses can be detected on the opposite sensor. For example, pulses transmitted from the right IR LED can be detected by the right, mid and even the left sensor as depicted below:

Right LED Pulses
IR Pulses from the oight LED may be detected on the left sensor. The degree of detection depends on the distance the obstacle is away from the robot. This data is used to confirm the proximity of an obstacle.

MHITS_R
IR Pulses from the right LED are detected by the Mid Sensor

LHITS_R
IR Pulses from th right LED are detected by the Left Sensor

RHITS_R
IR Pulses from the right LED are detected by the Right Sensor
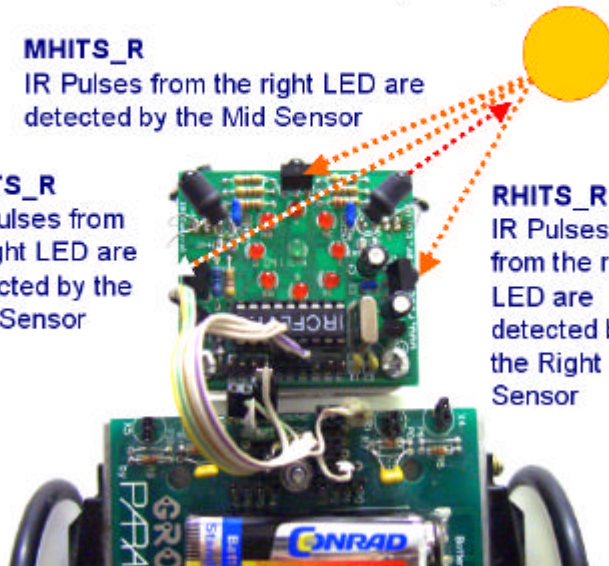
In the example above, the values of RHITS_R, MHITS_R, LHITS_R will increase inversely proportional to the distance of the obstacle. i.e. the closer the object the higher the value.

It is best to calibrate the **IRCF-L** by adjusting the Infrared LEDs in/out and up/down to get the best results for your application.

The main robot controller will decode data received from the **IRCF-L** and decide how to avoid the obstacle(s). The decision can be a Fuzzy logic algorithm based on which sensor is picking proximity data.

VERSION ICRF-L v2.0

Example Basic Stamp code:

```
GET_IR_CF_DATA_PACKET:
        ' Send control code 32 at @9600 baud
        SEROUT 2, 16468, [32]

        'Receive data stream form control freak
        'Put 6 Bytes of data into an array

        FOR DATA_ARRAY = 1 to 6
            SERIN 4, 16468,100,time_out,[RX(DATA_ARRAY)] ' Get 1 Byte
        NEXT

        'Display the 6 bytes of data on the debug screen
        FOR DATA_ARRAY = 1 to 6    ' Display the 6 bytes of data
            Debug DEC RX(DATA_ARRAY),"     ", CR
    ' Note, it is necessary to have two's loop otherwise data will be lost
        NEXT
RETURN
TIME_OUT:
        Debug "timed_out- Trying again", CR
        Goto Again

' Note, it is necessary to have two's loop otherwise data will be lost.

CHECK_IRPD:
        If RX(4) < 10 and RX(3) < 10 then skip2
        IF RX(4) =100 OR  RX(3) =100 THEN BACK  'Turn Back
        IF RX(4) =>99 AND RX(3) => 99 then CONDITION1   'Turn Back
        IF RX(3) > RX(4)  then CONDITION2               'Turn Left
        IF RX(4) > RX(3) then  CONDITION3               'Turn Right


RETURN
```

6.5   Command code 034 & 044 – Freewheeler mode

Syntax: 034 (decimal format)

Response from IR_CF: IRCF-L responds with 6-bytes of serial data (in decimal format), containing proximity information.. See command 044 for ASCII format.

This command works in the same way as command 032, expect that after being primed with one command code, the **IRCF-L** sends a continuous data stream of infrared proximity data in decimal format. The Serial cable (not power) can be disconnected and the module will keep working.

The only way to exit from this routine is to send another command code (such as 032).

Objective: This commands sends a continuous stream of proximity serial data in decimal format. It is used for testing the device and for calibration. It can also be used for non-robotic applications where a continuous stream of data may be required.

This command is useful for calibration and test purposes, when the **IRCF-L** is interfaced directly to a PC using an ASCII terminal such as Docklight, Stamplot, HyperTerminal, or other program. The ASCII output makes it also easy to interface to fuzzy logic calculating programs or for making graphical representations of the IRPD data in spreadsheets

You don't stop playing because you get old;...
...you get old because you stop playing!

6.6   Command Code 035 – Autopilot mode

Syntax: 035 (decimal format)

Response from IR_CF: [direction] [proximity] IRCF-L responds with 2-bytes of serial data, containing direction & proximity information.

Objective:
IRPD Mode Fuzzy logic results are sent as two bytes of data to serial port. These depict the direction of the nearest obstacle and distance.

This is similar to the command 032, but only returns two-bytes of information rather than 6 bytes. The main benefits of this command is that the decoding and interpretation has already been carried out within the IRCF-L based on the internal Fuzzy logic routines, so less calculation and processing is required by your robot controller.

The Circular display will also illuminate LEDs in the direction of the nearest obstacle.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!
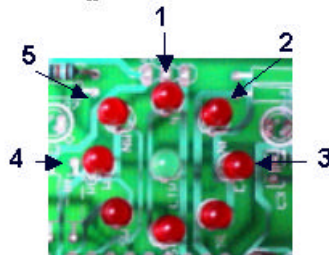
6.7 Command Code 036 – Infrared Beacon Mode

Syntax: 036 (decimal format)

Response from IR_CF: IRCF-L responds with 4-bytes of serial data.

Byte 1	Byte 2	Byte 3	Byte 4
Device Code	Button Code	Sensor Direction	No. of successful readings (0-5)

The LED closest to the IR sensor, in the circular LED display, will also illuminate.
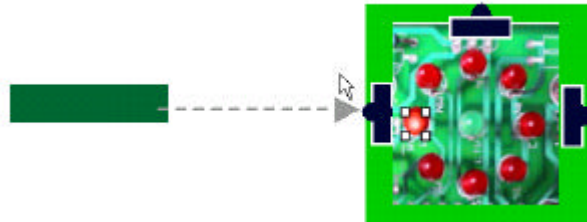
Example 1:
In the example below:

```
[TX] – 036
[RX] – 017 018 001 005
```

036 is transmitted to the IRCF-L module. The bytes returned are:
Byte 1 = device code = 017
Byte 2 = button code = 018
Byte 3 = The front sensor (North) detected highest hits (this is also the default direction of all sensors have the same number of hits)
Byte 4= 5 hits where received at the front sensor (North)

Example 2:
In the example below, no IR transmissions where detected:

```
[TX] – 036
[RX] – 000 000 000 000
```

Example 3:
In the example below the right sensor no. 3 (east) detected most IR hits:

```
[TX] – 036
[RX] – 017 018 003 005
```
The device code & Button code received was 017 and 018 respectively.

Example 4:
In the example below the left sensor no. 2 (west) detected most IR hits:

```
[TX] – 036
[RX] – 017 018 002 005
```

The device code & Button code received was 017 and 018 respectively.

Objective:
This command searches for Sony (SIRC) infrared transmission on all 3-sensors and returns with information on which sensor ID received the highest reading. There are a few different objectives of this command. The first method is for detecting the presence of other robots that are transmitting IR signals, for example in TAG games, Sumo competitions, etc. The second objective is to determine the location of an infrared beacon.

Setting up an Infrared Beacon
There are many ways to use the IRCF or IRCF-L as a beacon. One simple way is for a IRCF module (the Light version is not necessary for a beacon) to be used as a Beacon. This beacon will send out a continuous coded message (using code 052). The IRCF-L connected to the robot should then search for this message using command 030. The robot may roam around in a pattern then turn in a circle to try to locate the beacon. Once the robot picks up the message, the robot would send command 036 to IRCF-L to identify the direction of the infrared message.

Here you one IRCF modules to act as a beacon (The IRCF-L connected to the robot would send out a coded command in all directions trying to located the beacon. If the beacon picks up the command the beacon will acknowledge the request by sending a confirmation code Infrared command from a robot fitted with another IRCF-L module using command 020 and 030. The robot could send a command to a Beacon using command 020. As soon as one robot sends this command, the IRCF-L will respond by transmitting a device or button code representing a Beacon number. A robot fitted with an IRCF-L module would then identify the direction of the homing beacon by using command 036 (Beacon Mode).

Infrared Interference:
Take care when working in confined spaces, with white backgrounds, as the sensors will also pick up Infrared reflections. The actual direction may therefore be 180 degrees from the indicated direction. The robot controller should therefore run some check routines to verify these readings.

Reflected IR signal

Remote Control

wall
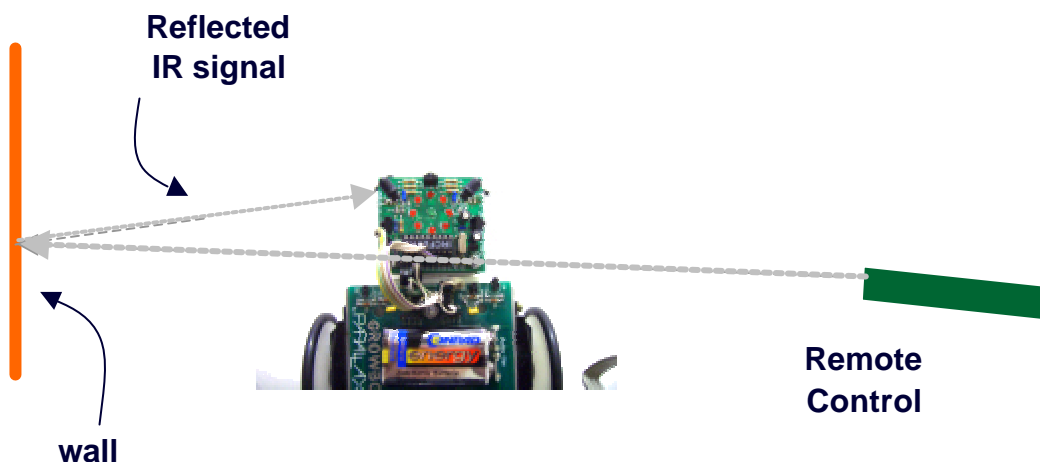
In Biomimetics simulation experiments, the IR transmitter could represent different food type using the device and button codes as type of smell.

**Robot
turns in the direction
of IR beacon**

IR coded beam

remote

You don't stop playing because you get old;...
...you get old because you stop playing!
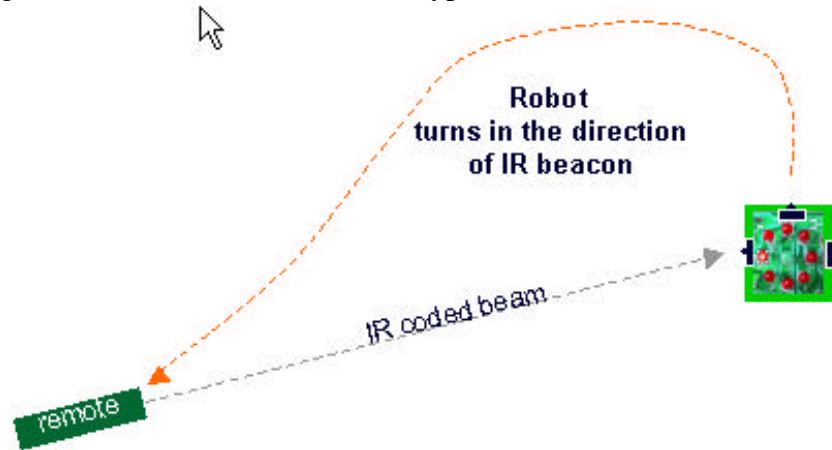
6.8 Command Code 037 - Light Sensor mode

Syntax: 037 (decimal format)

Response from IRCF-L: 2-bytes of data.

Byte 1	Byte 2
Right Sensor Light Level	Left Sensor Light Level

Objective:

This command measures the left and right sensor light levels and returns the light level value as two bytes of data in decimal format. The circular display also indicates the approx. direction of the brightest light source. With this command the robot can determine the direction of the brightest light source and head towards the brightest light source or hide away into the shadows. The following exercise is an example of how the IRCF-L: will search for the strongest light source and turn in the direction. See the website www.robotmaker.eu.com/ircf-l/examples/command_037/ for other examples and videos of how this command could be used.

Example:

```
'{$STAMP BS2}
'*************************************************
'*    037_Command_Light_Experiment_1.bs2        *
'*************************************************
'
' This is an exercise using the IRCF-L command 037 - Light Sensor Command.
' The Bot is programmed to follow the direction of light.
' The Bot will just simply turn in the direction of the brightest light.'
' See website http://www.robotmaker.eu.com for more details and video of experiment


'*************************************************
'*            GENERAL VARIABLES                 *
'*************************************************
  RX            VAR  Byte(6)
  DATA_ARRAY    VAR  Nib




' Mathematical variable
  LOOPCOUNT    VAR  Byte   'determine how far Growbot will backup and turn


'*************************************************
'*            CONTSTANTS                        *
'*************************************************
' Servo Speed Control on Boebot PCB
  MStop      CON 750  'Motor Stop
  Speed100   CON 125  ' Full Speed
  Speed075   CON 50   ' 3/4 speed
  Speed050   CON 40   ' 1/2 speed


'Servos
  R_SERVO  CON  14  'Pin for right servo
  L_SERVO  CON  13  'Pin for left servo

'LEDS
  L_LED   CON   15  ' Pin for the left LED
  R_LED   CON   11  ' Pin for the right LED
```

VERSION ICRF-L v2.0

```
'************************************************
'*                MAIN PROGRAM                 *
'************************************************
MAIN:
  GOSUB COMMAND_037
  GOSUB CHECK_LIGHT
GOTO MAIN

COMMAND_037:
  SEROUT  2, 16468, [37]
  SERIN  4, 16468,10,COMMAND_037,[RX(1),RX(2)]   ' Get 2 Bytes
  'DEBUG CR,DEC RX(1), "    ", DEC RX(2)  , CR
RETURN

CHECK_LIGHT:
  IF RX(1) > RX(2)  THEN RIGHT  'Turn Right
  IF RX(1) < RX(2)  THEN LEFT   'Turn Left
RETURN

RIGHT:
  LOW L_LED 'Turns off left LED
  HIGH R_LED 'Turns on right LED
  FOR LOOPCOUNT=1 TO 2
  PULSOUT R_SERVO,MSTOP-speed100   'Pulse servo
  PULSOUT L_SERVO,MSTOP-speed100   'Pulse servo
  PAUSE 20
  NEXT
RETURN
GOTO MAIN

LEFT:
  LOW R_LED 'Turns off left LED
  HIGH L_LED 'Turns on right LED
  FOR LOOPCOUNT=1 TO 2
  PULSOUT R_SERVO,MSTOP+speed100   ' Pulse servo
  PULSOUT L_SERVO,MSTOP+speed100   ' Pulse servo
  PAUSE 20
  NEXT
RETURN
GOTO MAIN
```

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

6.9   Command Code 038 - Light Sensor Mode + IRPD Mode

Syntax: 038 (decimal format)

Response from IRCF-L: 8-bytes of data.

RIGHT IR SENSORS		MIDDLE IR SENSOR		LEFT IR SENSOR	
BYTE1	BYTE2	BYTE3	BYTE4	BYTE5	BYTE6
RHITS_R Pulses detected by the right IR sensor that are transmitted from the RIGHT IR emitter.	RHITS_L Pulses detected by the right IR sensor that are transmitted from the LEFT IR emitter. The object will need to be very close for readings to be made.	MHITS_R Pulses detected by the middle IR sensor that are transmitted from the RIGHT emitter.	MHITS_L Pulses detected by the middle IR sensor that are transmitted from the LEFT IR emitter.	LHITS_L Pulses detected by the LEFT IR sensor that are transmitted from the LEFT IR emitter.	LHITS_R Pulses detected by the LEFT IR sensor, that are transmitted from the RIGHT IR emitter. The object will need to be very close for readings to be made.

RIGHT LIGHT SENSOR	LEFT LIGHT SENSOR
BYTE7	BYTE8
Right Sensor Light Level	Left Sensor Light Level

The Circular display will flash once in the direction of the nearest obstacle and once in the direction of the nearest light source for each command sent.

Objective:
This command is a combination of the light sensor command (037) and Infrared Proximity detection (IRPD) command (032). It would be quite normal that Infrared Proximity Detection (IRPD) to take priority over light sensing.. The objective of the command is to avoid crashing into an obstacle whilst heading towards (or away from) a light source.
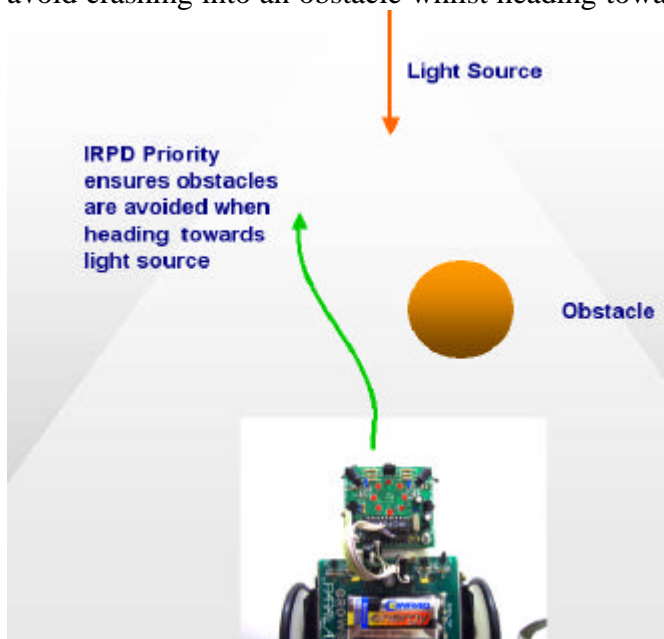
This command triggers the IRCF-L to return IRPD data and Light levels all in one command. This command also alleviates the need to send two separate commands to the IRCF-L; greatly reduces the processing time of the robot controller.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

Example:

```
'{$STAMP BS2}
'************************************************
'*     038_command_Light_IRPD_Example1.bs2        *
'************************************************
' This is an example of how you could to use the light sensor functionality of the IRCF-L
' using a GROWBOT/BOEBOT from Parallax INC, fitted with a Basic Stamp.
' The bot will move towards the brightest light. The Infrared proximity
' sensing takes priority over light sensing. The bot will therefore avoid the obstacle before
' trying to turn towards light.
' The program sends the command 038 to the IRCF-L.
' The IR-CF/L RETURN 8 bytes of proximity DATA
' 1=R_HITS R
' 2=R_HITS L
' 3=M_HITS_R
' 4=M_HITS_L
' 5=L_HITS_L
' 6=L_HITS_R
' 7= Right Light Sensor
' 8= Left Light Sensor
' If values on the mid-sensors are below 90, then there are no immediate dangers to avoid
' and the bot will move forward. At the same time the bot 'will turn in the direction of the
' strongest light source.
' See http://www.robotmaker.eu.com for more details and videos


'************************************************
'*             GENERAL VARIABLES              *
'************************************************
  RX          VAR  Byte(9)  'Setup array with 9 bytes of data
  DATA_ARRAY  VAR  Nib      'Values from 0-100
  BACK_COUNT  VAR  Nib      'Variable to detect when bot is trapped
  LOOPCOUNT   VAR  Byte     'determine how far Growbot will backup and turn


'************************************************
'*             CONTSTANTS                      *
'************************************************
'Servos
  R_SERVO  CON  14  'Pin for right servo
  L_SERVO  CON  13  'Pin for left servo

'Servo Control
  MStop       CON 750     'Motor Stop
  Speed100    CON 125     ' Full Speed
  Speed075    CON 50      ' 3/4 speed
  Speed050    CON 40      ' 1/2 speed

'LEDS on Growbot are on pin 15 and pin 11.
'Change this to suit LEDS on Boebot or other model.
  L_LED   CON   15  ' Pin for the left LED
  R_LED   CON   11  ' Pin for the right LED

'************************************************
'*             RESET VARIABLES                *
'************************************************
BACK_COUNT=0

'************************************************
'*             MAIN PROGRAM                    *
'************************************************
MAIN:
'RESET VARIABLES
FOR DATA_ARRAY = 1 TO 8
  RX(DATA_ARRAY)=0
NEXT
GOSUB COMMAND_038
GOSUB CHECK_NEXT_MOVE
GOSUB CRICKET_DISTANCE
GOTO MAIN

'************************************************
'*             SUBROUTINES                     *
'************************************************
COMMAND_038:
'8 bytes of data are then returned from the IRCF-L giving 180 degrees of
'proximity data and light sensor data.
```

```
SEROUT  2, 16468, [38]  ' Send control code 038 at @9600 baud
' Get 6 Byte
SERIN  4,  16468, 200, time_out, [RX(1), RX(2), RX(3), RX(4), RX(5), RX(6), RX(7), RX(8)]
'Display results on Debug Terminal
'FOR DATA_ARRAY = 1 TO 8
  'debug DEC RX(DATA_ARRAY)," ,"
  'debug CR
  'debug "_____", CR
'NEXT
RETURN

TIME_OUT:
'If data is not return from the IRCF-L within 200ms the routine is timed-out.
'debug "timed_out- Trying again", CR
GOTO MAIN

CHECK_NEXT_MOVE:
  IF RX(3) < 90 AND RX(4) < 90 THEN  CHECK_LIGHT    'No threat from Obstacle so Turn towards
Light
  IF RX(3) < 95 AND RX(4) < 95 THEN FORWARD_SLOW    'Move forward very slowly
  IF RX(4) > 97 OR  RX(3) > 97 THEN BACK            'Turn Back
  IF RX(3) > 90 AND RX(3)  > RX(4)  THEN RIGHT      'Turn Right
  IF RX(4) > 90 AND RX (4) > RX(3) THEN  LEFT       'Turn Left
RETURN

CHECK_LIGHT:
'This routine moves the bot forward and also adjust the bot in the direction of the brightest
light.
HIGH L_LED   'Turns on LED
HIGH R_LED   'Turns on LED
'The next loop controls the speed of the servos. A gradual acceleration is obtained
'By increasing the pulse width.
FOR LOOPCOUNT = 1 TO 100 STEP 10
  PULSOUT R_SERVO,MSTOP-speed050-loopcount   ' Pulse servo
  PULSOUT L_SERVO,MSTOP+speed050+loopcount   ' Pulse servo
  PAUSE 20 ' A long pause of 20ms, produces a smoother forward movement
NEXT
IF RX(7) > RX(8) THEN RIGHT_LIGHT       'Turn Right towards light
IF RX(8) > RX (7) THEN  LEFT_LIGHT      'Turn Left towards light
GOTO MAIN

RIGHT_LIGHT:
FOR LOOPCOUNT = 1 TO 5
GOSUB RIGHT
NEXT
RETURN

LEFT_LIGHT:
FOR LOOPCOUNT = 1 TO 5
GOSUB LEFT
NEXT
RETURN

FORWARD_SLOW:
IF RX(3) > 98 OR RX(4) > 98 THEN BACK
FOR LOOPCOUNT = 1 TO 2
  PULSOUT R_SERVO,MSTOP-speed050   'Pulse servo
  PULSOUT L_SERVO,MSTOP+speed050   'Pulse servo
  GOSUB CRICKET
  PAUSE 20
NEXT
GOTO MAIN

RIGHT:
LOW L_LED                             'Turns off LED
HIGH R_LED                            'Turns on LED
PULSOUT L_SERVO,MSTOP-speed100        ' Pulse servo
PULSOUT R_SERVO,MSTOP-speed100        ' Pulse servo
RETURN
GOTO MAIN

LEFT:
LOW R_LED                             'Turns off LED
HIGH L_LED                            'Turns on LED
PULSOUT L_SERVO,MSTOP+speed100        ' Pulse servo
PULSOUT R_SERVO,MSTOP+speed100        ' Pulse serv
```

```
RETURN
GOTO MAIN

TURN_AROUND:
'When the bot is 'trapped' in a corner the best solutions is to turn
'the bot 180 degrees around AND head in a different direction.
'This routine determines which direction to turn the bot
'It will also cancel the turn-around if there are obstacle close to the bot which would be '
' hit
IF RX(1) >95 OR RX(5) > 95 THEN skip3 ' if there are items left and right of the robot then
don't turn
IF RX(1)>= RX(5) THEN TURN_AROUND_R    'RHIT_R > LHITS_L
IF RX(5)>RX(1) THEN TURN_AROUND_L      'LHITS_L > RHITS_R
GOTO MAIN

SKIP3:
  'BACK_COUNT=0
RETURN

TURN_AROUND_L:
FOR LOOPCOUNT = 1 TO 200
  GOSUB LEFT
NEXT
BACK_COUNT=0
GOTO MAIN

TURN_AROUND_R:
LOW L_LED                           'Turns off LED
HIGH R_LED                          'Turns on LED
FOR LOOPCOUNT = 1 TO 200
  PULSOUT L_SERVO,MSTOP-speed100    ' Pulse servo
  PULSOUT R_SERVO,MSTOP-speed100    ' Pulse servo
NEXT
BACK_COUNT=0
GOTO MAIN

CRICKET_DISTANCE:                   'some interesting sounds
  FOR LOOPCOUNT =18  TO 12
  FREQOUT 8,LOOPCOUNT,RX(3)*rx(4)+2000
  'toggle spk
  NEXT
RETURN
```

This next routine will move the growbot/boebot backwards, if the left and right sensors detect obstacles ' on both sides. To avoid the robot being beginning locked in an endless back and forward loop, the number of times the robot needed to move backwards is recorded. If this routine is executed more than 5 times, the robot will just turn around and head in a different direction.

```
BACK:

HIGH L_LED                           'Turns on LED
HIGH R_LED                           'Turns on LED
FOR loopcount = 1 TO 5
  PULSOUT R_SERVO,MSTOP+speed100     'Pulse servo
  PULSOUT L_SERVO,MSTOP-speed100     'Pulse servo
  PAUSE 20
NEXT
BACK_COUNT=BACK_COUNT+1
IF BACK_COUNT >5 THEN TURN_AROUND
RETURN
```

You don't stop playing because you get old;...
...you get old because you stop playing!

6.10   Command Code 052 - Beacon Transmission Mode

Syntax: 052 <button Code><device Code> (decimal format)

Operands: <button Code><device Code>

Response from IR_CF: A letter X will flash on the circular display every 2 seconds when the IR pulse is transmitted.

Objective:
This command transmits a REPEATED Sony (SIRC) Infrared transmission command from both Infrared LED's. This commands works similar to command 020, but automatically re-transmits the IR device and button code every 2 seconds.

This command is intended to be used together with one or multiple IRCF / IRCF-L beacons, each sending different ID codes to a robot. The robots can then head for the correct beacon. It is also quite useful for debugging infrared devices, with the IRCF-L attached to a PC.

The IRCF-L can be disconnected from the serial connection after being 'primed' with a device code and a button code. It will then continue to keep transmitting the initial command every 2 seconds.

Any (SIRC) Device code + Button code can be transmitted.

Transmitting another command-codes to the IRCF-L will reset the module.

ROBOTMAKER
You don't stop playing because you get old;...
...you get old because you stop playing!

APPENDIX A – SAFETY, DISCLAIMER, LICENCE

We would like you to have fun with your Infrared Control Freak robot project, so please read and observe the safety warnings and disclaimer below carefully.

Young Technicians & Adult Supervision

Even though this kit is suitable for children over 14 years old, it is recommended that any soldering, testing and use of this module is carried our under adult supervision

Risk if Choking

The module contains small electronic parts, which may break-off, if it is mishandled, dropped, thrown, chewed. The small electronic parts that could fall off, may look like sweets to small children so there is a risk of choking if the small parts are swallowed. PLEASE keep all small parts away from small children, especially children under 5 years old.

Soldering Irons

Be very careful with the soldering iron! Grab it by the cool end (normally made of plastic) as the metal end gets hot enough to melt solder and will therefore also severely burn your skin if touched.

Please read and observer the manufactures instructions. It is assumed that you have done soldered before. If you want to read-up about how to solder, look at our website htttp://www.robotmaker.co.uk/ir_cf/soldering.htm.

To avoid fire hazard, remember to turn the soldering Iron OFF after use!

Safety goggles

To reduce the risk of eye accidents we strongly suggest you wear safety goggles when soldering or assembling parts to the device. Great care should be taken when cutting wire, as pieces can shoot off at great speed. Solder may also splash during soldering.
Always use safety glasses before you start soldering or working on the module.

Infrared light is not the same as Laser, however it is advisable not to look directly into the Infrared LED's when the unit is activated. Infrared light is not visible to the naked eye, so the eyes won't be able to react to intense IR light in the same way as natural light. Use a video camera with 'night vision' function, if you need to test the IR LED's are working correctly..

Antistatic

The Microprocessor on the IRCF-L module, can be easily damaged if you touch it without being grounded. To avoid static damage to the Microprocessor, you should ensure that you have connected an antistatic cable and have all components placed on an antistatic mat.

Correct Power Connection

When connecting power to the unit observe the maximum power limits, correct polarity and pin connections. Failure to connect correctly will destroy the Infrared Control Freak module.

VERSION ICRF-L v2.0

DISCLAIMER & LICENCE

To start with, please read the safety instructions (above) before doing anything! Please be careful and keep the IRCF-L module away from children and pets. The small electronic parts may break-off if it is dropped, thrown, chewed, etc..

If you are a child, make sure your parents know what you're doing. The infrared Control Freak is intended for hobby & educational robotic use for adults and also kids from the age of **14.** If you're under 14, you should ask an adult to help you and supervise you.

ROBOTmaker are not responsible if you hurt yourself or anyone else while assembling or using this module.

When interfacing the Infrared control Freak to your PC, robot controller, or any other device, it is your own responsibility to check that voltages and currents will not damage the equipment.

ROBOTmaker accepts no responsibility for damage or loss in any way.

By using this product, you agree not to hold ROBOTmaker or any other person liable for any injury or damages in any way.

When programming a robot remember to apply Asimov's Three Laws of Robotics:
1. Robots must never harm human beings or, through inaction, allow a human being to come to harm.
2. Robots must follow instructions from humans without violating rule 1.
3. Robots must protect themselves without violating the other rules.

The Infrared Control Freak is intended for hobby and educational use only. **Do not use it in ANY critical or life saving application!** Do not also use this product in any application where normal use or malfunction could cause injury or damage.

This manual in intended only as instructional value. Great care has been taken when compiling the information, however it cannot be guaranteed that any information is correct, accurate, or complete in any way and therefore no responsibility is taken for any errors or omissions.

No responsibility is also taken for any information on external links to websites.

None of the stated ambitions & claims are to be taken as a binding commitment. Anyone associated with ROBOTmaker does not assume any liability for damages resulting from the use of the information in this manual or take any responsibility for any infringement of intellectual property rights of third parties that would result from the use of this information. You use the information provided in this document at your own risk.

Copyright

IRCF/IRCF-L hardware and software design, including assembly manual, user manual, schematics and PCB layout are Copyright (C) 2003 ROBOTmaker. All rights reserved. If you want to use any of the material in this document or on the ROBOTmaker website, or if you are interested in distributing the IRCF-L, you need to contact support@robotmaker.co.uk and get permission in advance.

Please note that this document is not associated with or authorized by Parallax, Microchip, Lego or any other company mentioned in this document. This document has nothing to do with these companies or any other company in any way! Please contact support@robotmaker.co.uk if there any infringements and the section will be edited.

We would like you to have many hours fun with your robot project, so we ask you to take care and follow the safety warnings carefully.

Please ask if in doubt!

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

APPENDIX B – INTRODUCTION TO AUTOMONOMOUS ROBOTICS?

What is an autonomous robot?

An autonomous robot is simply an object that can move freely around on its own, with some degree of self-sufficiency, by reading sensors, following some programmed rules and learning from mistakes - without any human intervention.

The excitement and fun with robotics is really in the building and programming of the robot to cope with any new challenge it faces; whether just it's just roaming around your office or bedroom or coping with a more sophisticated challenge, such as in a game of robot tag or robot football and collaborating with other robots. There are many good websites with details about robots and robotics. Start by looking at http://robots.open.ac.uk/minicourse/ and http://www.realrobots.co.uk/Reading-uni.html

To get some ideas about robots in the future have a look at the videos of i-robot at http://www.cirg.reading.ac.uk/common/news/i_robot/making-i-robot-1-high.wmv

I'm a total beginner in Robotics. Where should I start?

You don't actually need much experience to get started with autonomous robotics. Robots can be bought as kits that can grow to suit your level of experience. It's very important that you are not put off by all the jargon and buzzwords and ensure your that you start at the right level.

You can pick and choose the areas that interest you most. For example the software student, who is primarily interested in learning to program in Java, C, Basic, etc can skip the electronics theory and just buy pre-assembled robot controller with sensors.

The same goes for mechanical students who may only feel comfortable with the mechanical assembly. The biological research student, interested in simulating artificial life, can skip a lot of technical background, go straight into high-level programming, and focus on simulation. There are many pre-built robot kits to help you get up-and-running very quickly.

If you are new to robotic building then we recommend that you start by buying a simple two-wheeled robot kit with the Infrared Control Freak - Light, to familiarise yourself with all the basic concepts and get to grips with some of the technical terms used.

ROBOTmaker manufacture a range of simple and advanced robot chassis's and stock may other types from various suppliers. Have a look at our website, where you'll find many places to buy robot kits and even how to build one from spare parts - that you may already have at home.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

IR-Control Freak LIGHT™ v2.0

Most of the simple autonomous robots have a single-chip microcontroller board and two wheels, powered by modified servos. These kits cost from about €50 to €200. For more advanced robotics there are many walking biped robots with two feet. The IRCF-L can easily connect to these robot controllers using a serial interface to provide a complete solution.

With the help of this manual and the examples on our website, you will be guided though various interesting experiments to help you understand the concepts of robotics and infrared proximity sensing (getting your robot to avoid hitting objects). We also cover some basic concepts of fuzzy logic and infrared remote control (control your robot from a TV controller).

What is a Robot Sensor?

Similar to the senses of human beings, animals and insect, it's just as important for a robot to be able to sense its environment, in order to survive. The robot will need to avoid hitting obstacles in a room or avoid falling down stairs. It may need to move towards light or hide in dark places. In an environment of multiple robots competitions, it will definitely need to communicate with other robots to recognise that is friend or foe.

Is the Infrared Control Freak - Light also suitable for beginners?

Yes! The Infrared Control Freak Light has been designed to get you 'up-and-running' with an autonomous robot project very quickly. It is suitable for all user groups - from novices to experienced robot makers; whether you are planning to build your 1st robot project or are designing a fuzzy logic sensor algorithms for your university Phd thesis. The IRCF-L is very easy to use with extremely powerful features.

Robotics is not just limited to the technology students or hobbyist! You can use robots as a simulation and research tool in many study areas - see some examples on our website http://www.robotmaker.co.uk/Index_research_development.htm of how robots are being used in Universities and research institutes as multidisciplinary modelling, simulation and learning tool in Biological, Math, Engineering and Technology. See also the paper Robots in Engineering from Southern Illinois University - Edwardsville.

You don't stop playing because you get old;...
...you get old because you stop playing!

APPENDIX C - Calibration of the Infrared Control Freak

As the infrared emitters and sensors can easily be moved out of position, it is recommended to calibrate the IRCF-L before initial use and after a period of continuous use. This is easily done by carefully bending the IR emitters and IR sensors to the correct positions. The calibration is carried out when the unit is connected to a PC or robot controller, using the continuous command 044. The Circular LED display will then indicate the direction of the nearest obstacle.

Start by placing the controller on a large black surface (e.g. card or paper) about 1-5cm off the ground, or ideally, at the exact height the module will normally be installed at.

Make sure there are not obstacles within 60cm range, then start sending the command 044. The circular display should now just show the default rotation display.

Place a white obstacle directly in-front (about matchbox size) and slowly bring this forward until the 'North' LED is illuminated but not the green centre LED. If necessary, adjust the left and right IR-Emitters so the North LED illuminates when the object is central. Bring the object closer and the green centre LED should illuminate.

Carryout the same exercise with the light sensors.

VERSION ICRF-L v2.0

APPENDIX D - Principles of 'Infrared Remote Control'

Infrared is an invisible light to the naked eye, with a wavelength of approximately 950nm. Infrared light however can be detected with one of the modern digital cameras that have night vision functionality.

Spectral sensitivity characteristics
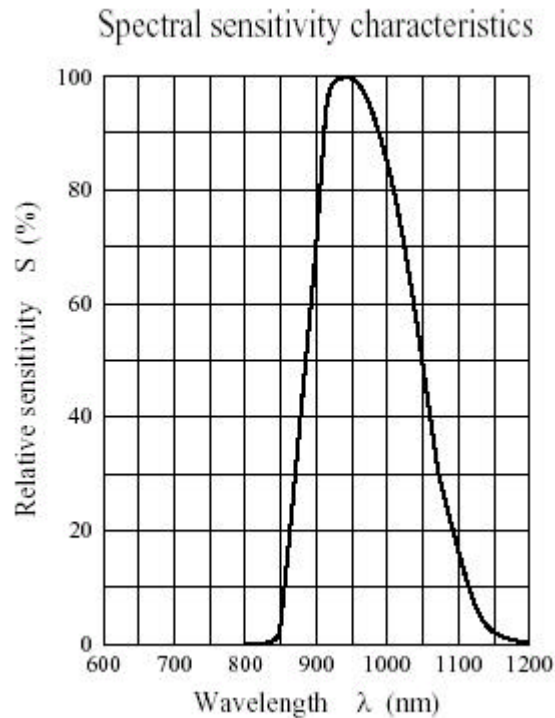
The SIRC protocol uses a pulse width encoding of the bits. This means that long and short pulses of infrared light are transmitted to create the digital signal. The code is similar to the principals of Morse code using long and short sound pulses. A long pulse of 1.2ms is represented a digital 1 and a short pulse of 0.6ms is represented as a logic 0. There is a 0.6ms pause between each pulse.
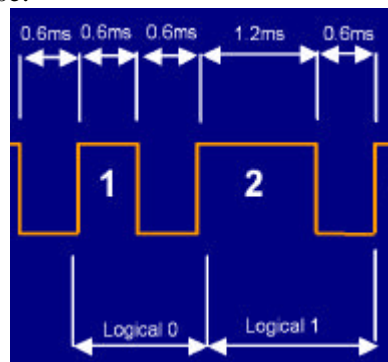
To gain some immunity from natural Infrared light (e.g. sunlight) or ambient light sources, the IR sensors 'trigger' only when a 32khz-40khz modulated IR signal is sent.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

IR-Control Freak LIGHT™ v2.0
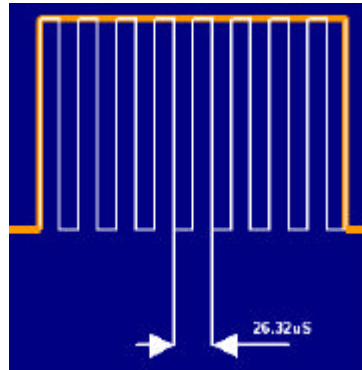
This means that , the sensors will only recognise a 1 or 0 pulse, when the infrared source is sending bursts of Infrared light at a frequency of 32Khz – 40Khz. If we use the formulae [seconds=1/frequency], we see that the pulse width is 31,25– 25.0 uS (microseconds). Remember that uS is one millionth of a second!

The sensors used in the **IRCF-L** are PNA4620M-ND from Panasonic (or similar component), which has a centre frequency of 38Khz. The sensitivity of these sensors is dependent on the modulated frequency. The frequency bandwidths are very narrow. A few Khz too low or too high reduces the sensitivity dramatically, as depicted in the graph below.
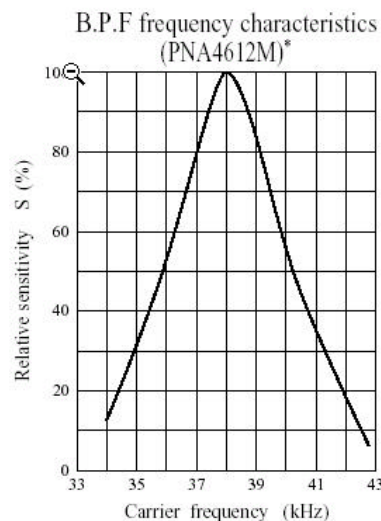
The SIRC infrared signal is made up of a 12-bit packet, which is split into a 7-byte button code and a 5-byte device code.

Start	BUTTON CODE							DEVICE CODE				
S	B1	B2	B3	B4	B5	B6	B6	D1	D2	D3	D4	D5
2.4ms	1.2 mS or 0.6mS							1.2 mS or 0.6mS				

The button code represents the actual button pressed on the remote control. There is a 2.4ms 'start bit' between each packet, which is used to synchronize the sensors.
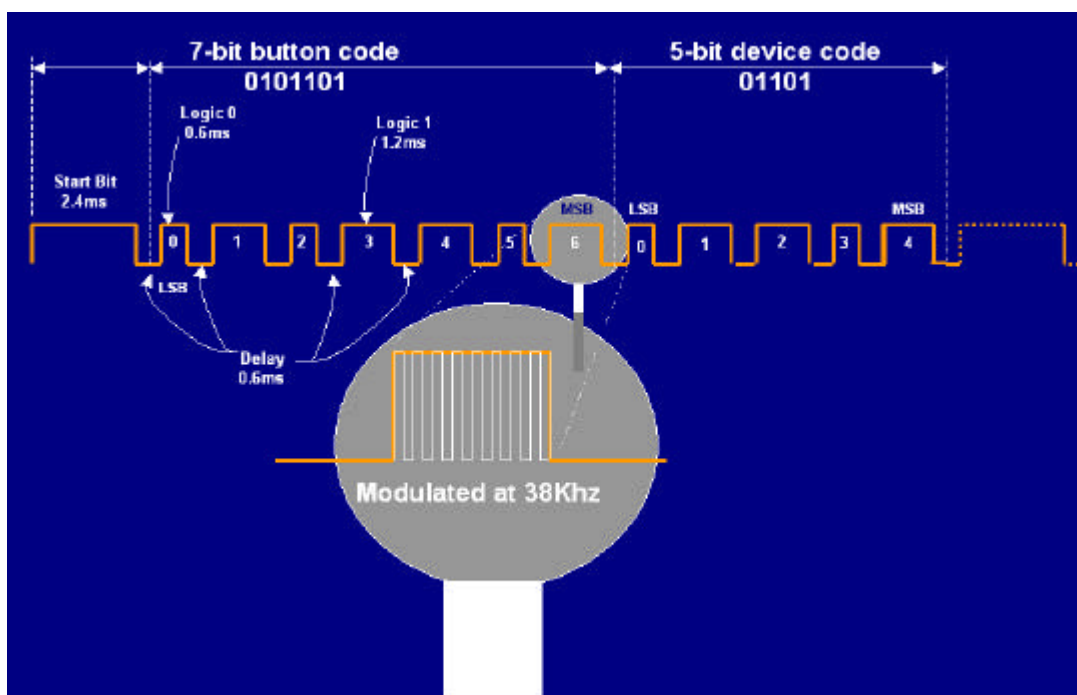
The device code determines which remote control device is being address; such as video, television, CD, Amplifier, SAT, etc. There are many documented versions of what these device codes actually represent. Here are some examples:

Address	Device	Device	Address
TV	1	TV	1
VTR1	2	VCR 1	2
Text	3	VCR 2	3
Widescreen	4	Laser Disc Unit	6
MDP	6	Surround Sound	12
VTR2	7	Cassette deck / Tuner	16
VTR3	11	CD Player	17
Effect	12	Equaliser	18
Audio	16		
Pro-Logic	18		
DVD	26		

Likewise, there are many documented versions of button codes. It best to test your own IR remote controller and create your own list. Here are some examples:

Button	Command	Button	Command
0	Digit key 1	20	Mute
1	Digit key 2	21	Power
2	Digit key 3	22	Reset
3	Digit key 4	23	Audio Mode
4	Digit key 5	24	Contrast +
5	Digit key 6	25	Contrast -

VERSION ICRF-L v2.0

6	Digit key 7	26	Colour +
7	Digit key 8	27	Colour -
8	Digit key 9	30	Brightness +
9	Digit key 0	31	Brightness -
16	Channel +	38	Balance Left
17	Channel -	39	Balance Right
18	Volume +	47	Standby
19	Volume -		

APPENDIX E – A BRIEF OVERVIEW OF FUZZY LOGIC

I don't intend to explain the theory of Fuzzy logic here. As you'll see below, there are already many websites and books on the subject of fuzzy logic, fuzzy control or fuzzy set theory. I merely would like to show the potential of creating your own fuzzy logic rule sets using the Infrared Control freak-Light.

Fuzzy Logic was initiated in 1965 by Lotfi A. Zadeh, professor for computer science at the University of California in Berkeley, as a means to model the uncertainty of natural language.

Zadeh says that rather than regarding fuzzy theory as a single theory, we should regard the process of ``fuzzification'' as a methodology to generalize ANY specific theory from a crisp (discrete) to a continuous (fuzzy) form (see "extension principle" in [2]). Thus recently researchers have also introduced "fuzzy calculus", "fuzzy differential equations", and so on.

Fuzzy Logic can be described basically as a multi-valued logic that allows intermediate values to be defined between conventional evaluations like yes/no, true/false, black/white, etc. Notions like rather warm or pretty cold can be formulated mathematically and processed by computers.

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth -- truth-values between "completely true" and "completely false".

Charles Elkan, an assistant professor of computer science and engineering at the University of California at San Diego, offers the following definition:

 "Fuzzy logic is a generalization of standard logic, in which a concept can possess a degree of truth anywhere between 0.0 and 1.0. Standard logic applies only to concepts that are completely true (having degree of truth 1.0) or completely false (having degree of truth 0.0). Fuzzy logic is supposed to be used for reasoning about inherently vague concepts, such as 'tallness.' For example, we might say that 'President Clinton is tall,' with degree of truth of 0.9...."

Building FL from the IRCF-L commands
The best command to use for to build your own Fuzzy Logic rule sets is the command 032; where six bytes of proximity information is returned for each command sent. Using the feedback from IRCF-L. the same fuzzy definitions could apply for the definition on the distance of an object. When is an object very near, near, far or very far away? As an example, the values on the Infrared Control freak could be mapped to something like this
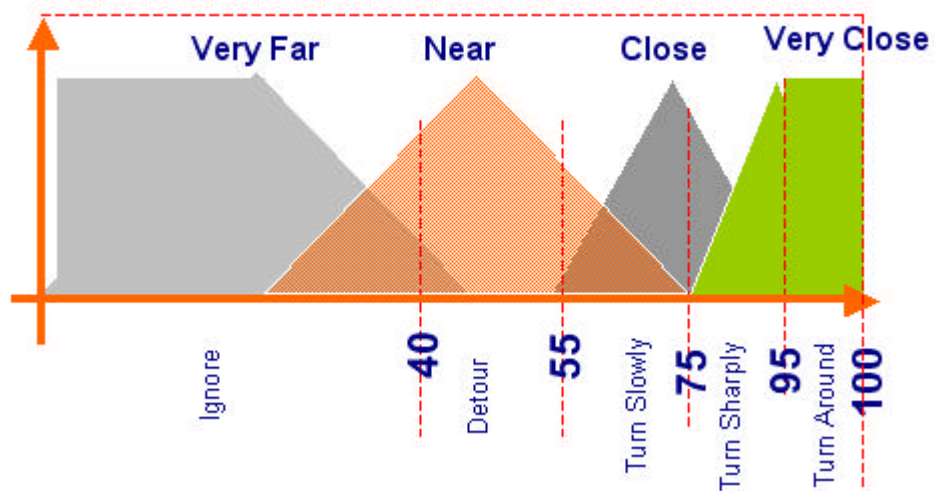
Usually fuzzy logic variables are also mapped with another measurement such as 'angle of object' in the case proximity sensing. The angle of the object can be roughly determined by mapping measurements from two or more sensors; similar to the follow diagram.
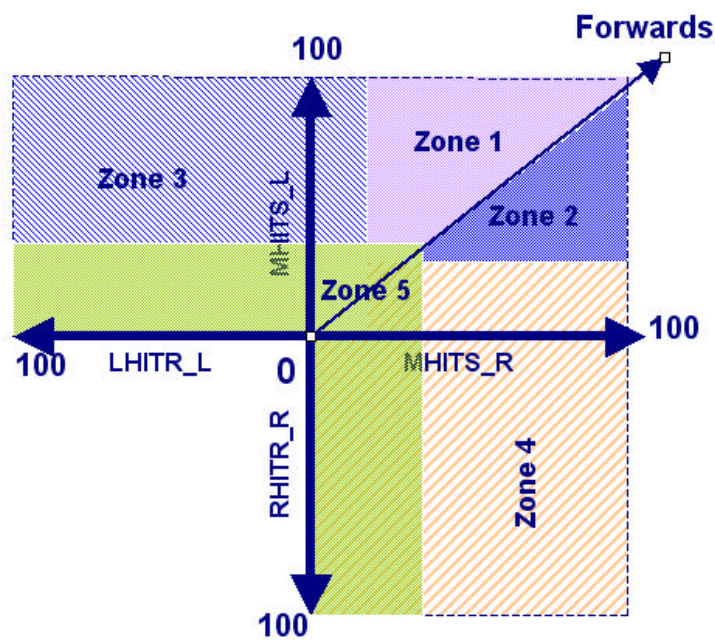
The fuzzy logic variables correspond in fact to a multidimensional model, The one above is 3 dimensional. If rate of change of obstacle were added that would then be the fourth dimension.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

90°
45°
90°
45°
45°
90°
45°
90°

100

MHITS_L=75
MHITS_R=75
A

MHITS_R=75
MHITS_L=25
C₁

MHITS_R=75
RHITS_R=75
C₂

Measurement on Centre Sensor (MHITS_R)

Obstacle 'B' at 45°

MHITS_L=75
MHITS_R=25
B₁

Measurement on Centre IR sensor (MHITS_L)

Measurement on Right IR Sensor (RHITS_R)

100

100

MHITS_L=75
LHITS_L=75
B₂

Measurement on Left Sensor (LHITS_L)

100

45°

90°

45°

90°

=No Action
=Detour
=Turn Slow
=Turn Fast

A

C

C₁
C₂
45°

B
B₁
B₂

B

You don't stop playing because you get old;...
...you get old because you stop playing!

IR-Control Freak LIGHT™ v2.0

Now we have several rules that say what to do in certain situations:

Consider, for example, that an obstacle (in relation to the IRCF-L) is dead head (also defined as example A in diagram above).
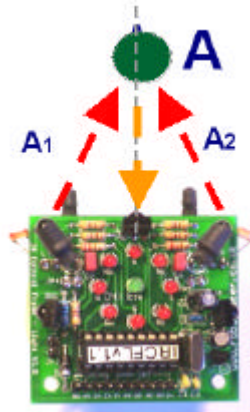
The angle is zero. Both A1 and A2 give equal readings on the MHITS_L and MHITS_R Obviously if this were fitted to a robot, this is the un-desired situation, and therefore the robot would need to stop or turn around to avoid hitting the obstacle.

Let's consider another case: where the obstacle is towards the Left. In this case the robot may need to avoid the obstacle by turning slightly to the right.
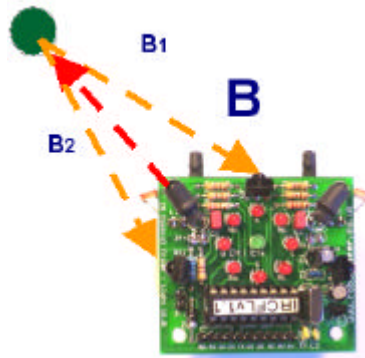
Alternatively if the rate of change in reading were measured (dy/dt) then it could be confirmed whether the obstacle was moving towards the sensor or not, or just passing by.

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

IR-Control Freak LIGHT™ v2.0

Measurement

The relative distance to an object decreases as the robot moves past an obstacle (distance c->b), By tracking the direction it is possible to determine whether the object is moving or stationary and whether it is a 'threat' or not
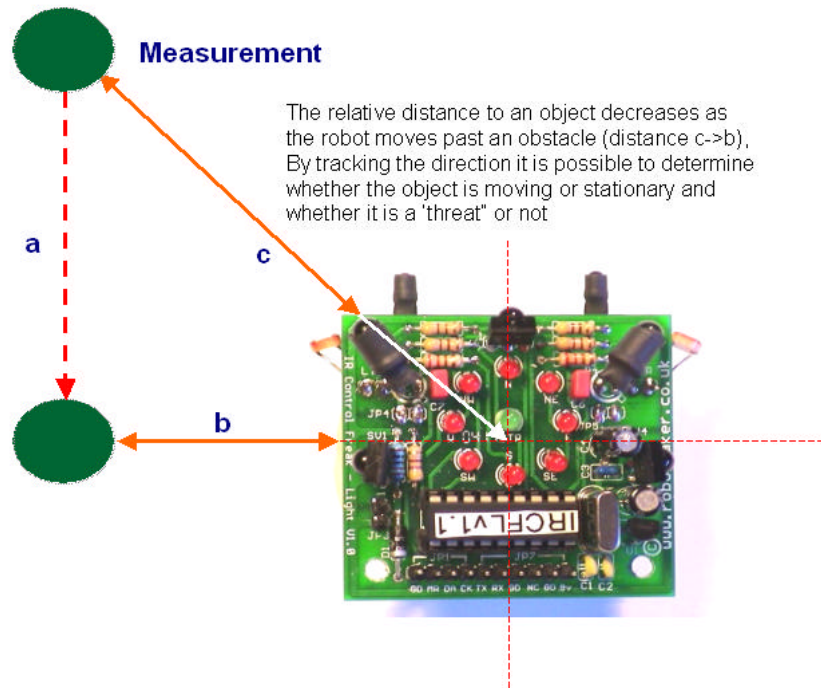
So far two rules have been made-up that can be put into a more formalized form like this:
- ?? If angle is zero and MHITS_L AND MHITS_R >90 then speed = NEG HIGH on left motor and NEG HIGH one right motor.
- ?? If angle is 45 degrees and MHITS_L and RHITS_L = 75 then speed = POSITIVE LOW on left motor and ZERO on right motor.

Where:
- ?? NEGATIVE -HIGH = reverse motor full power
- ?? NEGATIVE -LOW = reverse motor low power
- ?? ZERO = Motor off
- ?? POSITIVE LOW = forward motor low power
- ?? POSITIVE HIGH = Forward motor full power

Defuzzification methods

There are many ways to *defuzzy* the problem. The table below shows how the left and right motor speed we can summarize all applicable rules in a table:

Proximity	Very close		Close		Near		Far		Very far	
	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
Zone 1	NH	NH	PL	NL	PL	PL	PH	PH	PH	PH
Zone 2	PL	NL	Z	PL	PL	PL	PH	PH	PH	PH
Zone 3	Z	PL	PL	Z	PL	PL	PH	PH	PH	PH
Zone 4	PH	PL	PL	PL	PL	PL	PH	PH	PH	PH
Zone 5	PH	PH	PH	PH	PL	PL	PH	PH	PH	PH

NH=Negative High, NL=Negative Low, Z=Zero. PL=Positive Low, PH=Positive High
The voltage on the motors could vary depending on the degree of Negative or Positive values.

References:

- ?? *A Brief Course in Fuzzy Logic and Fuzzy Control*. By Peter Bauer, Stephan Nouak, and Roman Winkler, and offered by the Fuzzy Logic Laboratorium Linz - Hagenberg.
- ?? See also list of applications of fuzzy logic. http://www.flll.uni-linz.ac.at/navigation/main_navigation/frame_aboutus.html

- ?? The Article Fuzzy Logic-An Introduction by Steven D. Kaehler on the Seattle Robotics Society web site: http://www.seattlerobotics.org/encoder/mar98/fuz/fl_part1.html
- ?? Mobile Robot Control Using Fuzzy Logic Vellasco et al.
- ?? The comp.ai.fuzzy newsgroup:http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/html/faqs/ai/fuzzy/part1/faq-doc-2.html

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

APPENDIX F – CIRCUIT DIAGRAM

VERSION ICRF-L v2.0

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!
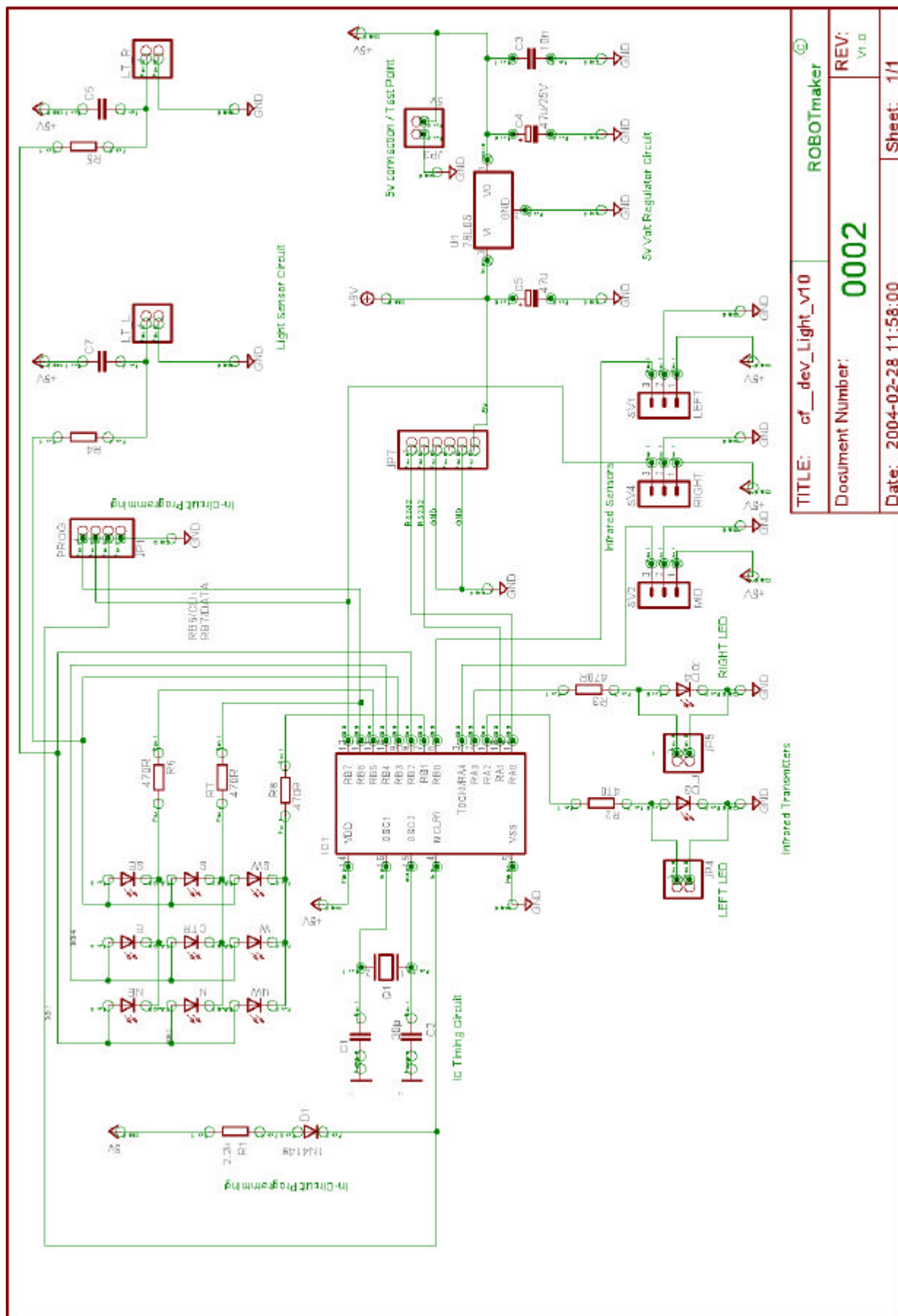
APPENDIX G – PCB LAYOUT

Layout of components
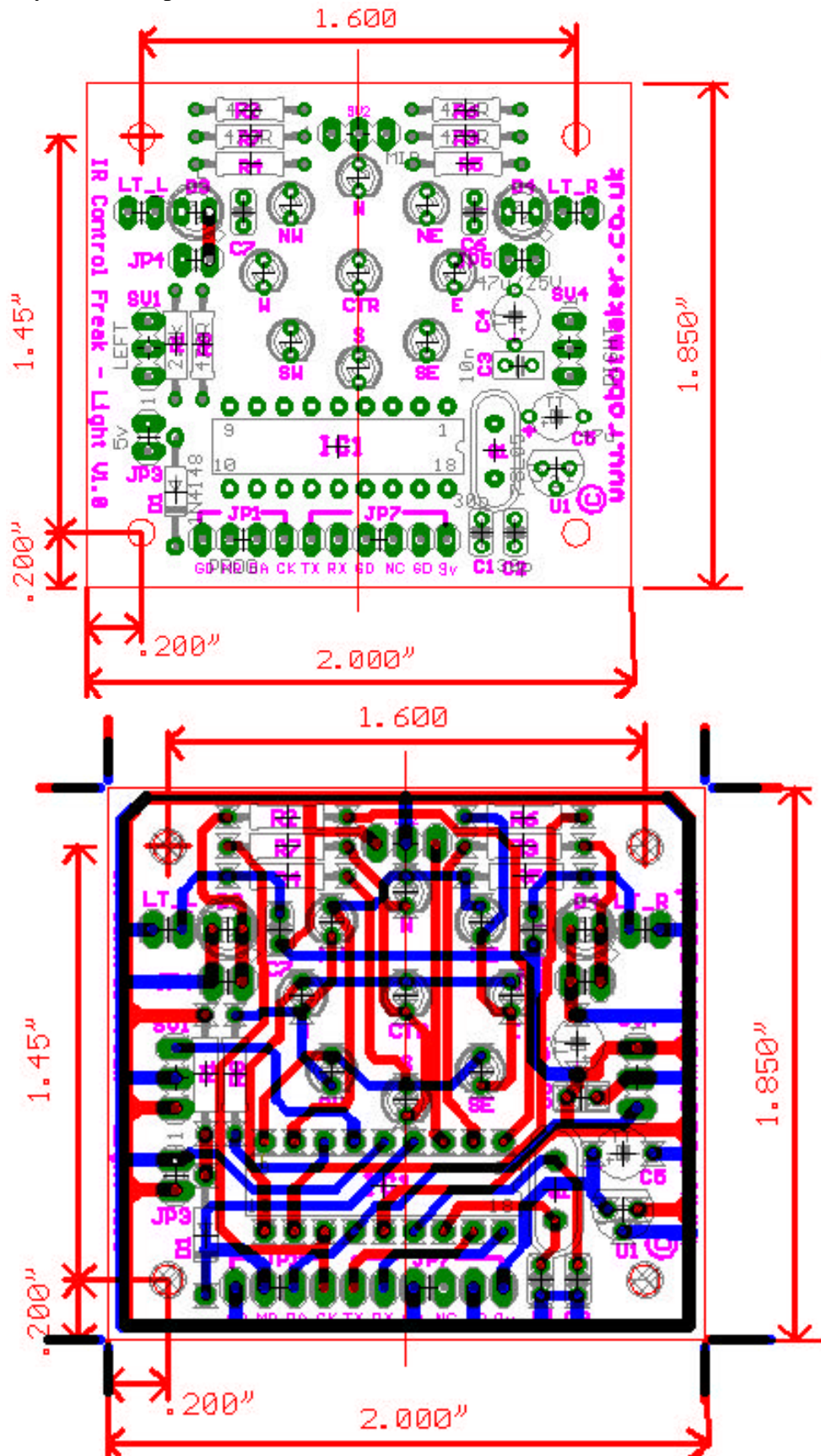
APPENDIX H – FAULT DIAGNOSTICS

If you have any questions or improvement suggestions related to the product, please feel free to leave a message in the support forum: http://www.robotmaker.co.uk/phpbb   or send an email to: support@robotmaker.co.uk

Fault Description 1: - No power at the test point JMP1.
1. Check battery /supply voltage is present on Pin 1.
2. Check ground connections
3. If these are OK, check for short circuits
4. Check polarity of battery
5. Check the voltage regulator is inserted correct way round (applicable to kit only).

Fault Description 2: - Circular LED display Start-up LED routine does not work
1. Check fault description 1.
2. Check that the IC is inserted into the socket correctly
3. Check that power is available at the IC
4. Check that the LEDs are connected the correct way round (applicable to kit only)

Fault Description 2: - No sensing or response from IRCF-L
If the power-up routine or other command mode routines are displayed on the circular display, but the module does not react to any proximity or IR signal.
1. Check battery is fully charged. The sensors will not trigger if the voltage is too low.
2. The Microcontroller will start behaving erratically if the voltage is too low.

VERSION ICRF-L v2.0

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

VERSION ICRF-L v2.0

APPENDIX I - Connector Overview

| Pin No.	IC Socket Pin	IC PIN Name	Description
1	NA Regulated 5v supply to pin 5	NA Regulated 5v supply VDD	9V SUPPLY (or regulated 5v supply from other source).
2	5	VSS	GROUND – for power supply
3	5	VSS	GROUND – for serial interface
4	18	RA1	Serial connection RX - Pin A1 on PIC. This is connected to the data transmit (TX) on the main robot controller
5	17	RA0	Serial connection TX – Pin A0 on PIC This is connected to the data receive (RX) on the main robot controller
JP1			
1	12	RB6	CLK
2	13	RB7	DATA
3	4	MCLR	MCLR
4	5	VSS	GROUND

VERSION ICRF-L v2.0

APPENDIX J - Modifications for increased range.

The IRCF-L has been designed to provide optimum sensor distance for most robot projects. However, in a few circumstances, an increased sensing range may be required. We recommend that you purchase the Infrared Control Freak POWER[?] range of infrared sensors that are designed with longer range sensing functionality (production versions will be available from begin 2005).

For students with some electronics understanding and who have done soldering before, it is possible to achieve an increased range by modifying the IRCF-L to achieve an extend range of about **1 meter** (1yard).

We have observed that due to background "noise" i.e. infrared reflections from walls, ceiling and the ground, the IRCF-L module becomes somewhat over-sensitive and may not give reliable IRPD information with this modification. The noise effect is greatly dependent on how the IRPD module is mounted on the robot. Higher mounting reduces ground effect noise. Noise is also reduced by using dark backgrounds (e.g., black ground and walls).

The range can be modified by inserting additional Infrared LED's and by experimenting with different IR LED's voltage limiting resistor values.
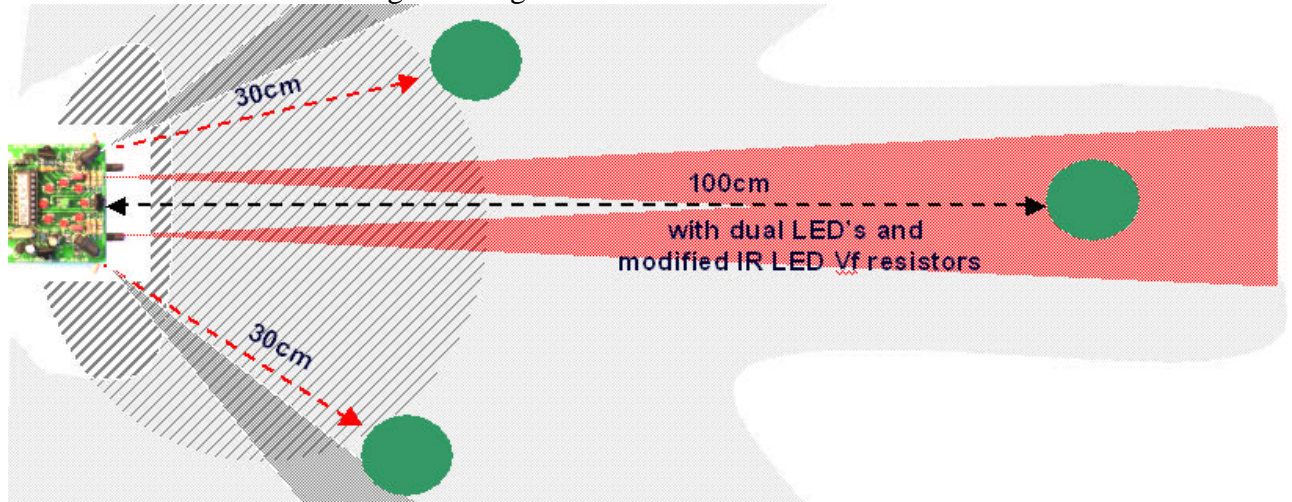
Currently, 470-ohm resistors are used to give the optimal distance. To obtain maximum distance sensing, replace the 470-ohm resistors R2 and R3 with 150-ohm resistors (or insert 220-ohm resistor in parallel with the existing 470-ohm resistor). If you are using two sets of infrared LED's, then replace R2 & R3 with 100-ohm resistors (or 127-ohm resistor in parallel with the existing 470-Ohm resistors). See http://www.electronics2000.co.uk/calc/calcled.htm for more details of LED limiting resistor calculations.

Care should be taken **NOT** to increase the forward current to more than 20mA otherwise the Microcontroller will "fry". Any modifications are done at your own risk!

ROBOTMAKER

You don't stop playing because you get old;...
...you get old because you stop playing!

APPENDIX K – GRAPHICAL REPRESENTATION OF INFRARED PROXIMITY DATA

It is possible to graphically interpret the infrared data.

Building your own program, you can download a simple version of Microsoft Visual Studio from the Microsoft WebPages.

You should covert your own fuzzy logic routines into a graphically representations. Depict an obstacle based on the data that is collected. You will probably need to filter and smooth out the data to avoid to erratic changes.

To get you going an example visual basic program (and source code) is available. The IR Control Freak GUI was a tool for testing and providing graphical representation of proximity data transmitted from the IR control freak module.

The program is still in an early development phase, so anyone is welcome to continue developing the tool further. The software is not support and is provided just for information purposes.

The software application is written in Microsoft Visual Basic and incorporates the COMM-library from Willies Computer Software Co. (see below). The COMM drivers interface from your PC's Comm Port (e.g. COMM1) to your program. At your own risk you can download time limited trail version of the drivers from:

Willies Computer Software Co. Website: http://www.wcscnet.com/CdrvLBroStandard.htm
Drivers Download: http://www.wcscnet.com/ftp/demo/cdrvl190.exe

In case the application has been maliciously tampered with, it would be advisable to always check for viruses before executing any downloaded software.

The IRCF-L GUI can be downloaded from www.robotmaker.co.uk/IRCF-L /gui/

This program is an only a rough example of graphical representation of proximity data being transmitted. A lot of work still needs to be done. Download and install at your own risk. The Software is not supported in any way.

RobotMaker

You don't stop playing because you get old;...
...you get old because you stop playing!
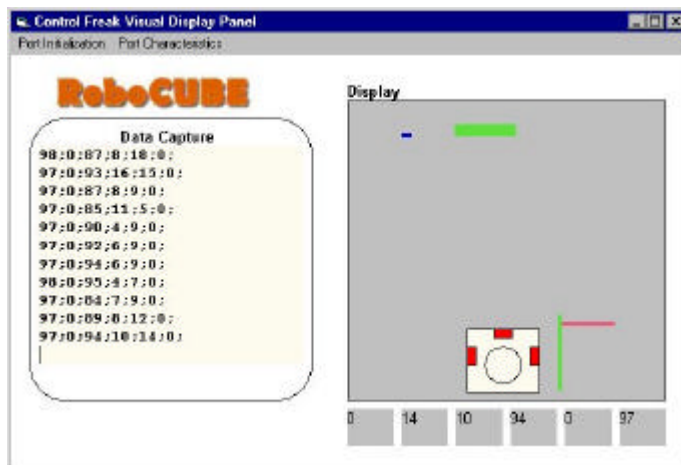
APPENDIX J - ASCII TABLES

(See: www.asciitable.com for more details.)

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	Y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|		
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.asciitable.com

128	Ç	144	É	160	á	176	▒	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	161	í	177	▓	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	162	ó	178	█	195	├	211	╙	227	π	243	≤
131	â	147	ô	163	ú	179	│	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	164	ñ	180	┤	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	165	Ñ	181	╡	198	╞	214	╓	230	µ	246	÷
134	å	150	û	166	ª	182	╢	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	167	º	183	╖	200	╚	216	╪	232	Φ	248	°
136	ê	152	_	168	¿	184	╕	201	╔	217	┘	233	Θ	249	∙
137	ë	153	Ö	169	⌐	185	╣	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	170	¬	186	║	203	╦	219	█	235	δ	251	√
139	ï	156	£	171	½	187	╗	204	╠	220	▄	236	∞	252	ⁿ
140	î	157	¥	172	¼	188	╝	205	=	221	▌	237	φ	253	²
141	ì	158		173	¡	189	╜	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	174	«	190	╛	207	╧	223	▀	239	∩	255	
143	Å	192	└	175	»	191	┐	208	╨	224	α	240	≡		

Source: www.asciitable.com