XC800 Family

AP08121

Infrared Remote Controller with Capacitive Touch Interface

Application Note

V1.0 2011-02

Microcontrollers

Edition 2011-02

Published by
Infineon Technologies AG
81726 Munich, Germany

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

XC82x
Revision History: V1.0, 2011-02

Previous Version: none

Page	Subjects (major changes since last revision)

1 Introduction

In today's Human-Machine Interface (HMI) designs, touch buttons, touch sliders and wheels are all important input elements. The use of capacitive touch technology to create these elements gives flexibility, a high-level of customization and a significant reduction in overall system cost.

The design of mechanical buttons is complex and costly. Cut-off clearances must fulfill certain requirements in terms of thermal expansion and aesthetic perception, while the material of the buttons and the electric contacts must be flexible, reliable and persistent. All these factors must be taken into consideration in the design, along with the incorporation of other elements such as potentiometers, to define the final size and shape of input buttons.

In contrast, the designer of capacitive touch systems can focus on interface requirements, taking advantage of the flexible and ultra-flat design solutions that the technology offers, and the ability to detect user input through enclosure materials, adjusting behaviour simply through software parameters rather than being dependent on mechanical constraints.

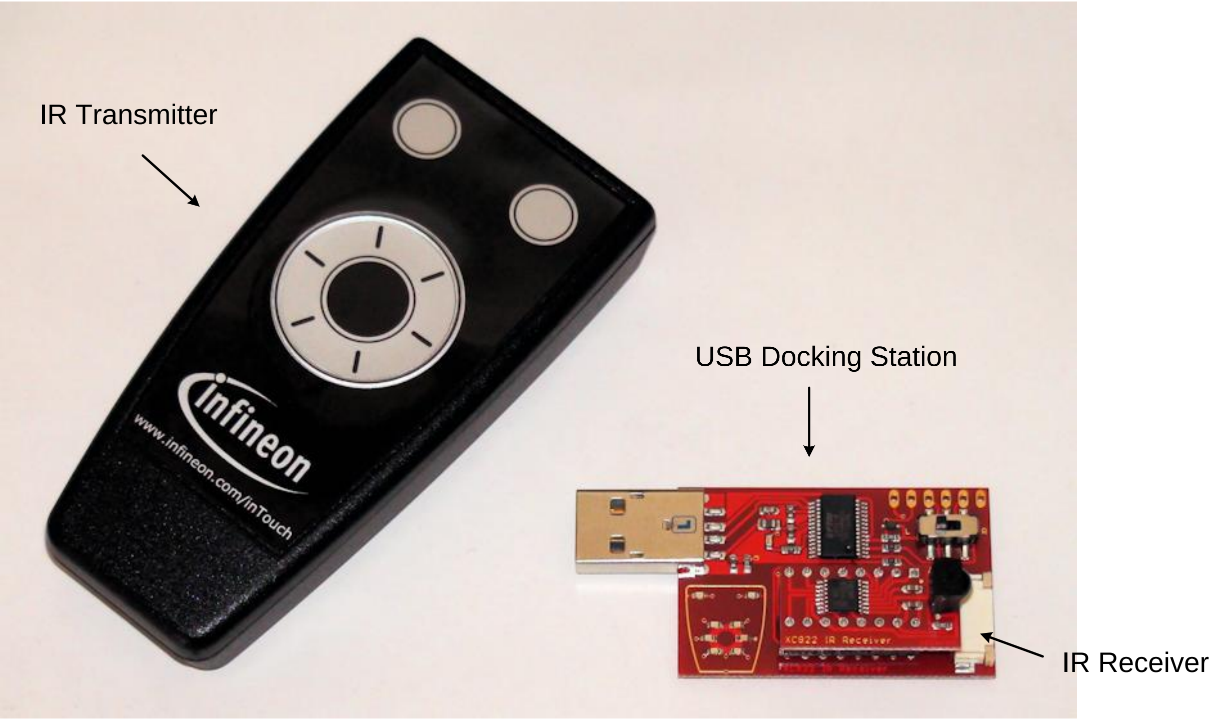


Figure 1 IR Remote Control Kit

The IR Remote Control Kit provides all the components necessary to evaluate capacitive touch techniques in infrared (IR) remote control solutions.

The kit provides all necessary components for evaluation of capacitive touch technique in infrared (IR) remote control solutions. The remote controller is an infrared transmitter which contains two touch buttons and a touch wheel. The infrared receiver board is in DIP16 form factor and can be used in many custom evaluation systems. The USB docking station provides 8 signaling LEDs and a flash programming interface for both the receiver and the transmitter. With the infrared receiver in the docking station, the 8 LEDs act as a simple display.

The embedded software for both the receiver and transmitter is part of the kit. Together with the signal processing of the touch sense unit, the standard RC-5 protocol is implemented for infrared transmission and reception. The transmitter software additionally includes a power-down and wake-up sequence to fulfill low power requirements.

The tool chain *DAVE™ Bench* is based on Eclipse technology and can be downloaded free of charge from http://www.infineon.com/dave-bench. It includes a compiler, a flash loader, a debugger, an IDE and a real-time user interface called U-Spy.

2 Hardware Setup and Basic Program Flow

2.1 IR Transmitter

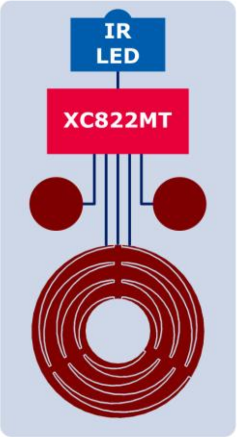


Figure 2 IR Transmitter

The *IR Transmitter* is on a flexible PCB which is glued directly onto the upper part of the housing. As a result, the touch elements are in close contact with the cover material with minimum air gaps. The infrared LED, a reverse shining SMD type, is mounted together with the 8-bit microcontroller XC822MT at the same flexible PCB. This part is folded into the front of the housing.

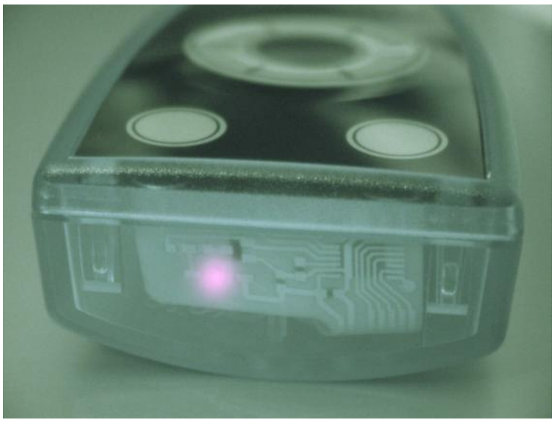


Figure 3 IR Transmitter viewed by a camera in night vision mode

Users can tap or touch-and-hold the two Infrared Transmitter buttons, and they can tap or dial the touch wheel.

The infrared diode is directly controlled by the 8-bit microcontroller XC822MT which generates a 36 kHz carrier wave and modulates the bit-stream to be transmitted.

The 3V power supply of the transmitter is realized by two AAA batteries. The flexible PCB is connected to this power supply, but can be removed and plugged into the USB docking station for flash programming or UART communication.

2.2 IR Receiver

The IR Receiver device consists of an infrared receiver IC combined with the XC822MT microcontroller. Next to the power supply pins (2.9V to 5.5V), just one additional pin is required for the infrared receiver IC. The remaining 12 IOs can be used for custom application signals.

The received signal is decoded using the T2 module of XC822.

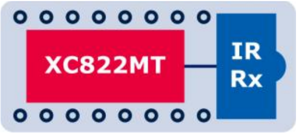


Figure 4 IR Receiver

2.3 USB Docking Station

The receiver board is plugged into a *USB Docking Station* which provides 5V power supply, a USB-to-UART Bridge and eight signaling LEDs. The LEDs can be driven by toggling IO pins or by PWM signals from XC822MT.

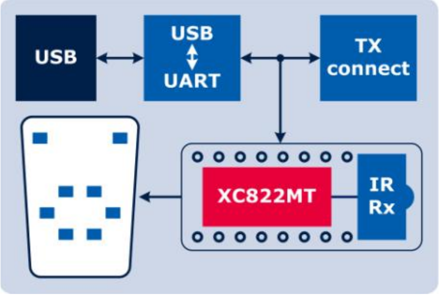


Figure 5 USB Docking Station

The default software of the kit uses the capture/compare unit CCU6 to dim a pair of LEDs according to the touch wheel’s position to mimic human dialing. This is done by making use of the multi-channel mode of CCU6.

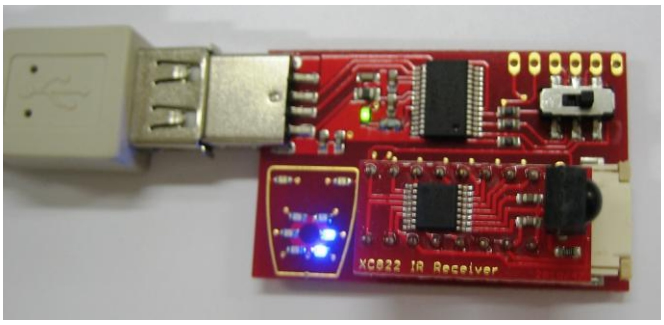


Figure 6 USB Docking Station displaying human dialing

2.4 Program Flow

IR Transmitter

The microcontroller spends most of the time in idle or power-down mode to limit power consumption. Touch sense related tasks are performed with high priority each time pad capacitances have been measured. Infrared transmission has a lower priority because of its very low speed, while UART communication has the lowest priority because it is not time-critical.

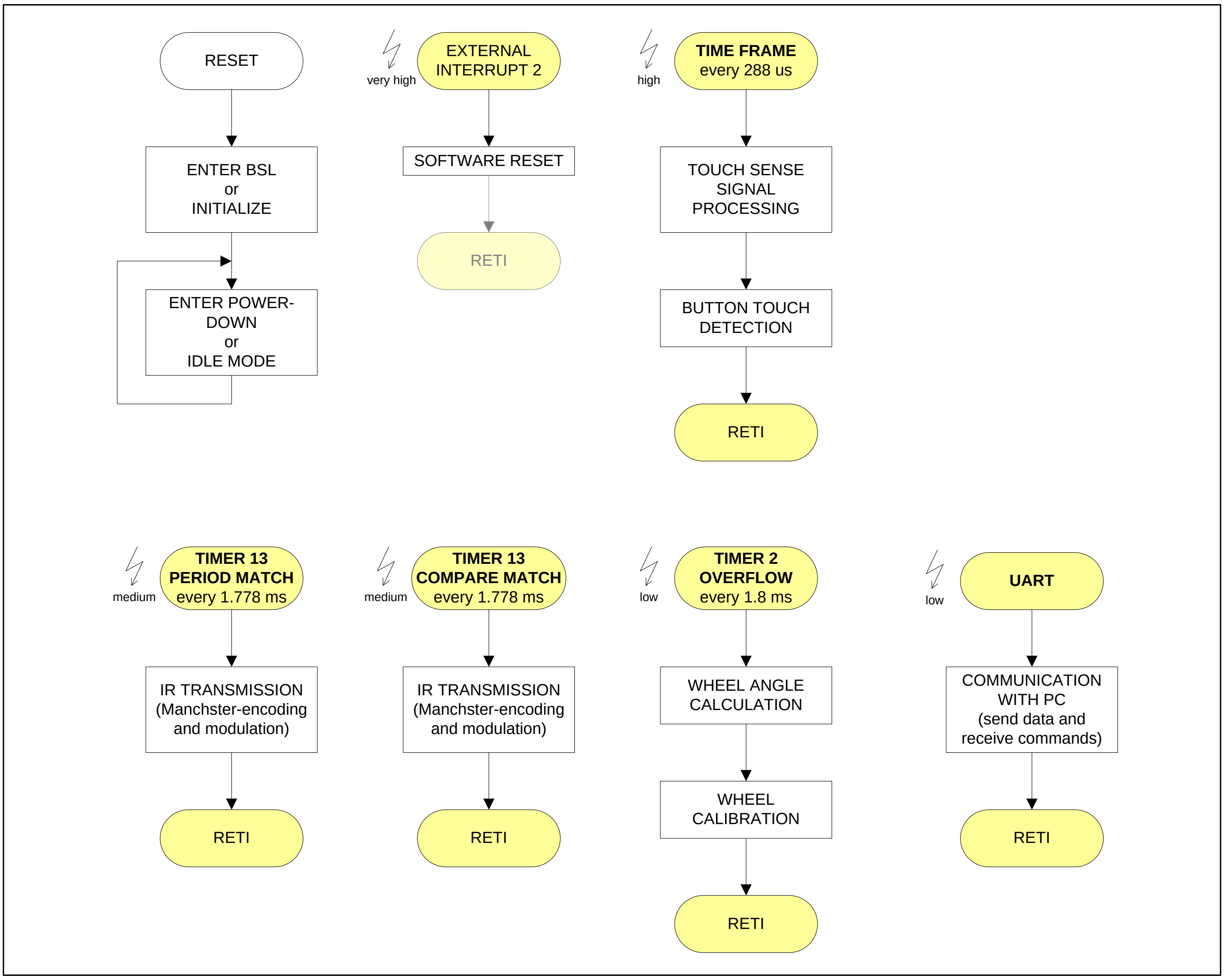

RESET
ENTER BSL or INITIALIZE
ENTER POWER-DOWN or IDLE MODE
very high
EXTERNAL INTERRUPT 2
SOFTWARE RESET
RETI
high
TIME FRAME every 288 us
TOUCH SENSE SIGNAL PROCESSING
BUTTON TOUCH DETECTION
RETI
medium
TIMER 13 PERIOD MATCH every 1.778 ms
IR TRANSMISSION (Manchster-encoding and modulation)
RETI
medium
TIMER 13 COMPARE MATCH every 1.778 ms
IR TRANSMISSION (Manchster-encoding and modulation)
RETI
low
TIMER 2 OVERFLOW every 1.8 ms
WHEEL ANGLE CALCULATION
WHEEL CALIBRATION
RETI
low
UART
COMMUNICATION WITH PC (send data and receive commands)
RETI

Figure 7 Overview of the main functions and interrupts of the IR Transmitter

Only a few of the XC822MT modules are required to perform all the required functions, with minimal CPU-load.

- The LED Touch-Sense Control Unit (*LEDTSCU*) handles the touch pads; it generates a so-called *Time Frame* interrupt after every measurement where signal processing and touch detection take place
- The Capture/Compare Unit 6 (*CCU6*) drives the infrared LED for IR and generates the *Timer 13 Period Match* and the *Timer 13 Compare Match* interrupts
- The Timer 2 (*T2*) module provides a slow time base by generating the *Timer 2 Overflow* interrupt for calculations necessary to handle the touch wheel
- The UART module, which is part of the XC800 core, is used for full-duplex UART communication with the PC

IR Receiver

Communication with the RC-5 protocol is very slow, so it is not critical to set interrupt priorities, although for high-speed communication (e.g. RF protocols) it could be critical.

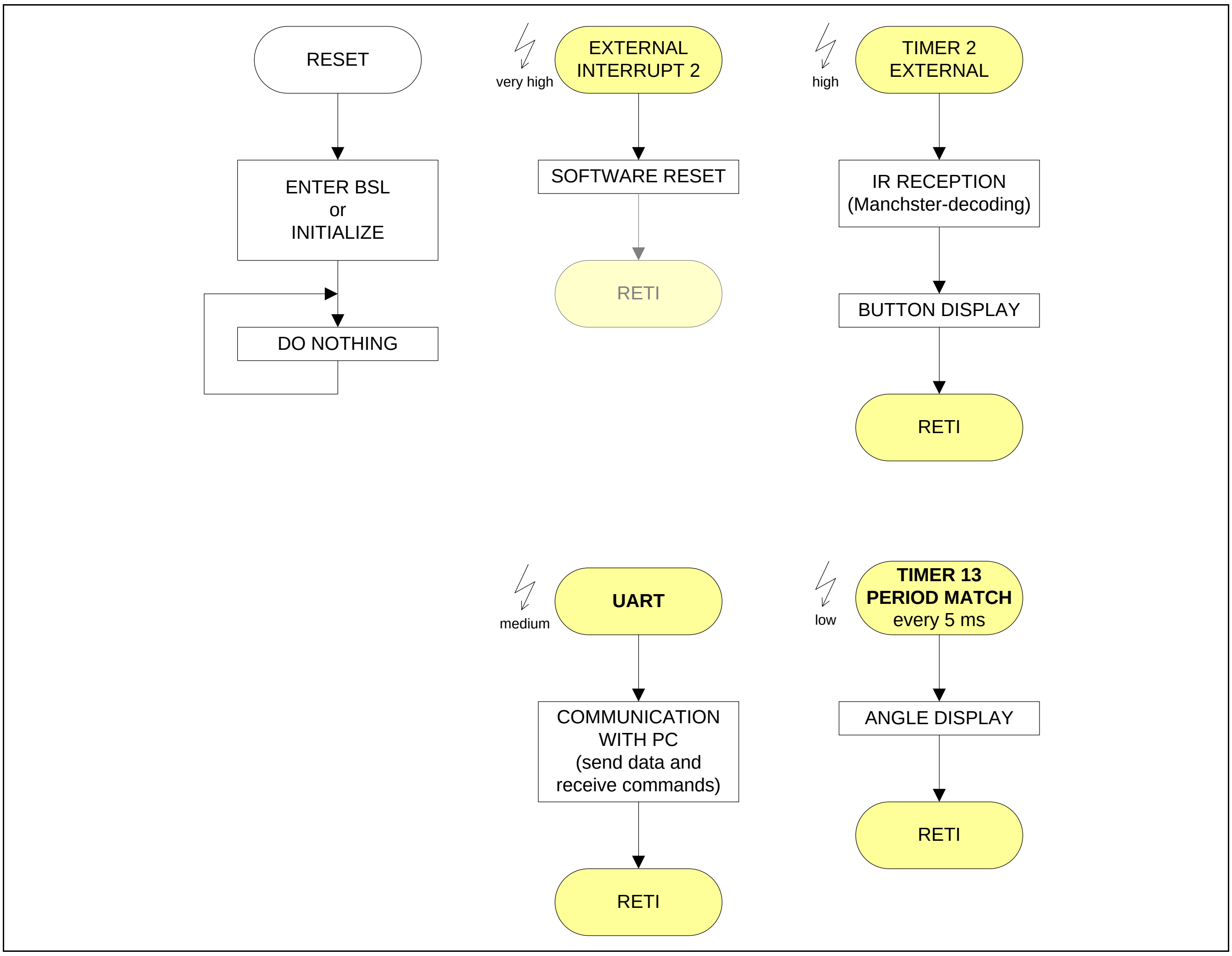


Figure 8 Overview of the main functions and interrupts of the IR Receiver

Modules used:

- Timer 2 (*T2*) takes care of receiving and decoding the infrared signal and it generates a *Timer 2 External* interrupt every time there is a change in the incoming signal level
- The Capture/Compare Unit 6 (*CCU6*) adjusts the brightness of the six wheel LEDs on the *USB Docking Station* and provides a slow time base by the *Timer 13 Period Match* interrupt for brightness level calculations
- The UART module, which is part of the XC800 core, is used for full-duplex UART communication with the PC

3 Infrared Communication

Most audio and video players can be controlled with infrared remote controllers. The infrared (IR) spectrum has less ambient noise than visible light which makes it very suitable for simple and reliable communication.

The typical setup includes an IR transmitter, an IR receiver and a protocol for communication. One of the most widely used protocols is the RC-5 standard from Philips. The communication is one way only; the transmitter sends commands to the receiver. The transmitter does this by toggling the on-board infrared LED (TSML1000) on and off. The infrared light of the LED is detected by a photo diode in the receiver IC (TSOP34836) on the receiver board.

3.1 Protocol

The transmitter follows the RC-5 protocol and the LED is toggled on/off at a frequency of 36 kHz, the so-called carrier frequency. The carrier frequency in turn is modulated by a bit stream at a much lower frequency. The bitstream itself is Manchester-encoded by the transmitting microcontroller. Manchester-encoding, or bi-phase modulation, means that every bit is divided into two chips, 0 and 1. Bit 1 contains a 0-1 chip transition while bit 0 contains a 1-0 chip-transition. Manchester code ensures that there is at least one line voltage transition per bit, which helps recovery. The receiver IC de-modulates the received signal, but decoding is handled by the receiving microcontroller. TSOP34836 inverts the signal.

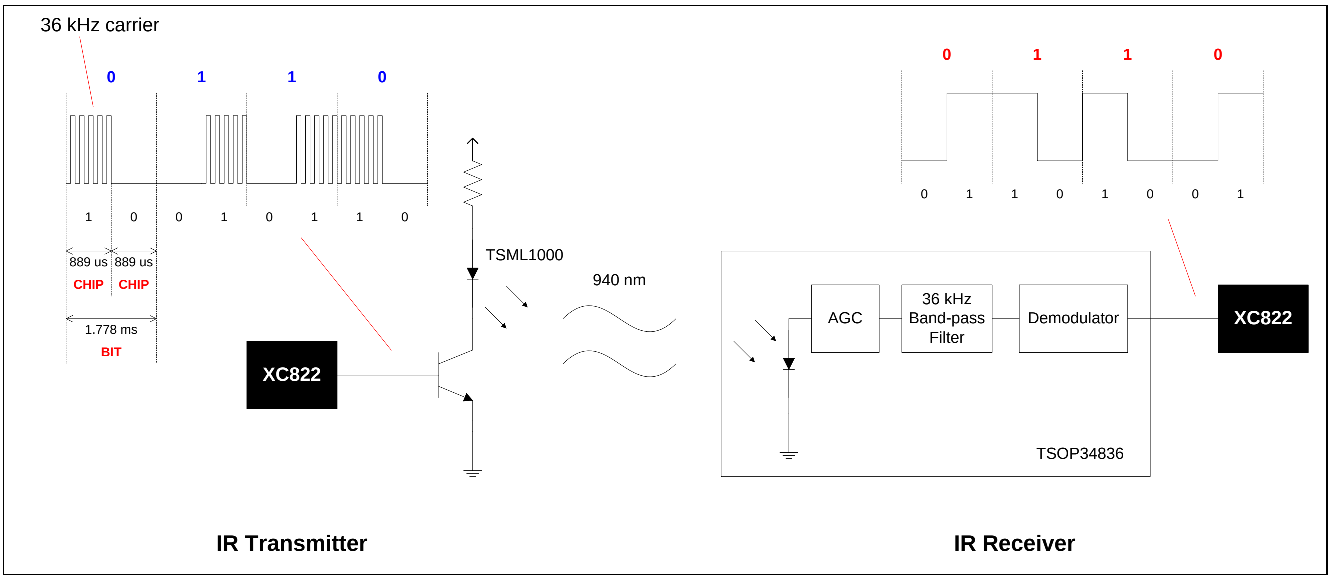


Figure 9 Infrared Transmitter and Receiver

By default, when the *IR Transmitter* board is powered by batteries (3V), the 36 kHz carrier wave has a 25% duty cycle to reduce power consumption but still have enough transmitting power for acceptable range. If the *IR Transmitter* board is plugged into the docking station and is powered by the USB (5V), the duty cycle is reduced to 5% to compensate for the higher diode current. In both cases there are exactly 32 pulses in a chip.

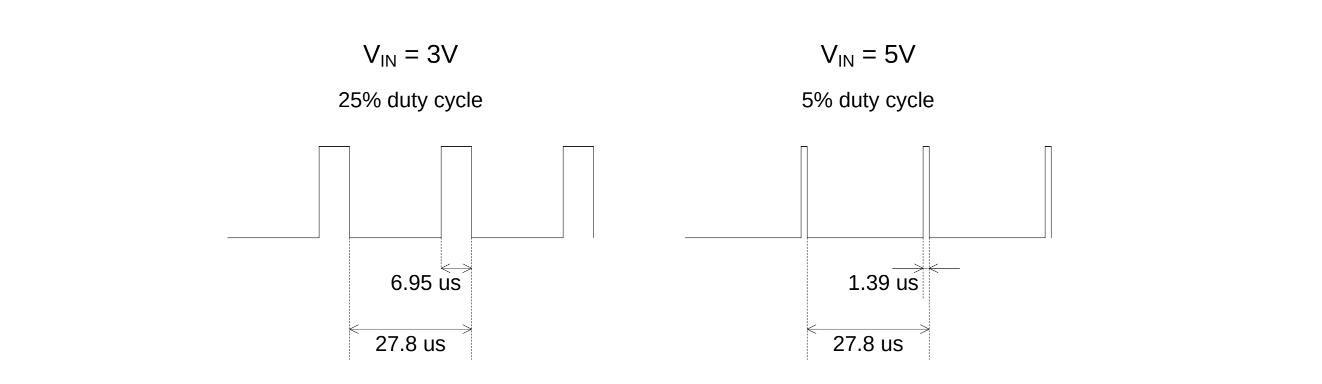


Figure 10 36 kHz carrier wave at 3V and 5V supply voltage

The RC-5 protocol uses 14-bit packets for transmission. The packets are transmitted one after another with a break of at least 89.1 ms between them. Each packet takes 24.9 ms to transmit.

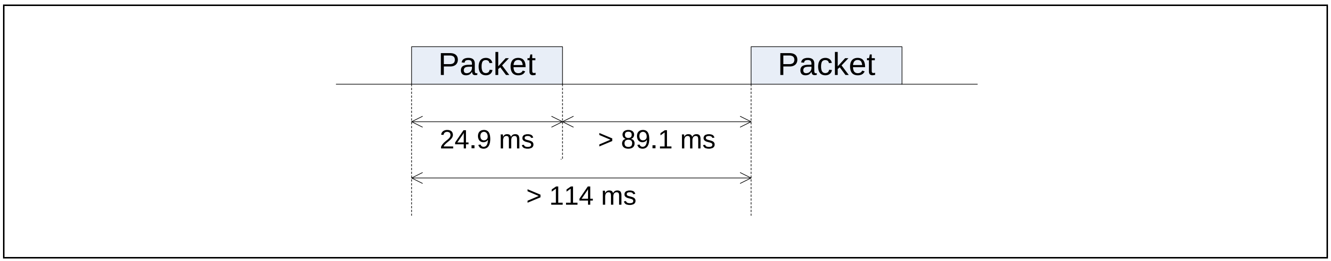


Figure 11 Transmission of packets

One packet contains the following bits and bit fields (in order of transmission):

- Start Bit (**S1** or S): 1 bit, always logic 1
- Field Bit (**S2** or F): 1 bit, can be used as command extension, always logic 1 in the *IR Remote Kit*
- Toggle Bit (**T** or C): 1 bit, toggles if new data is sent, does not change if the same data is sent repeatedly (e.g. a button is continuously touched)
- Address (**A4:A0**): 5 bits, MSB first, selects the target device, the *IR Remote Kit* does not use the standard RC-5 addresses, the *Receiver* board has multiple addresses instead
  - **01000**: Touch buttons and calibration
  - **11XXX**: Touch wheel
- Data (**D5:D0**): 6 bits, MSB first, one of the possible 64 commands, the *IR Remote Kit* does not use the standard RC-5 commands and this bitfield is used as general data (e.g. wheel section angle)

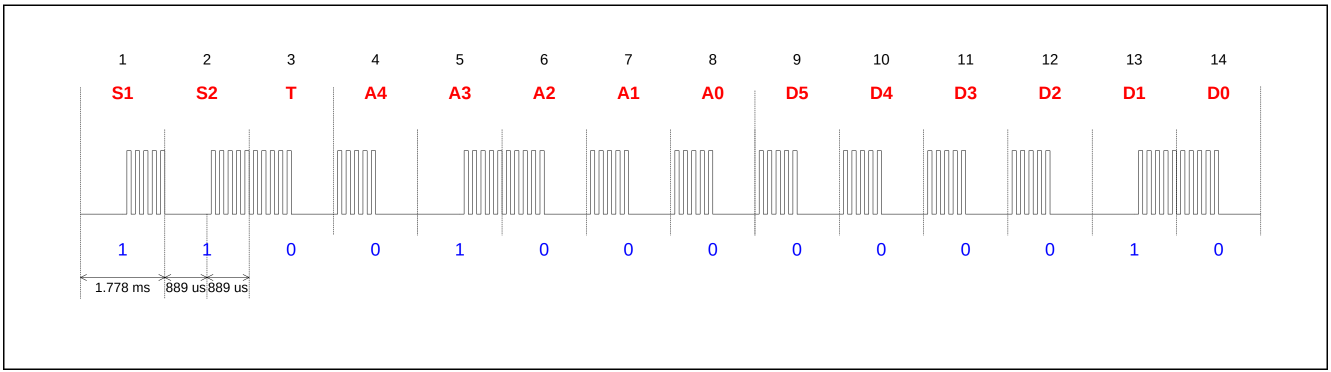


Figure 12 Bits of a typical packet

The *Receiver* board has a separate address for the buttons and calibration. In this case, the RC-5 address is **01000**, the Toggle Bit and [D2:D0] data bits are also used.

- D2 is used as Calibration Bit (**C**): 1 if the *Transmitter* board is self-calibrating
- D1 is used as Right Button (**BR**): 1 if the right button is touched, 0 if untouched
- D0 is used as Left Button (**BL**): 1 if the left button is touched, 0 if untouched

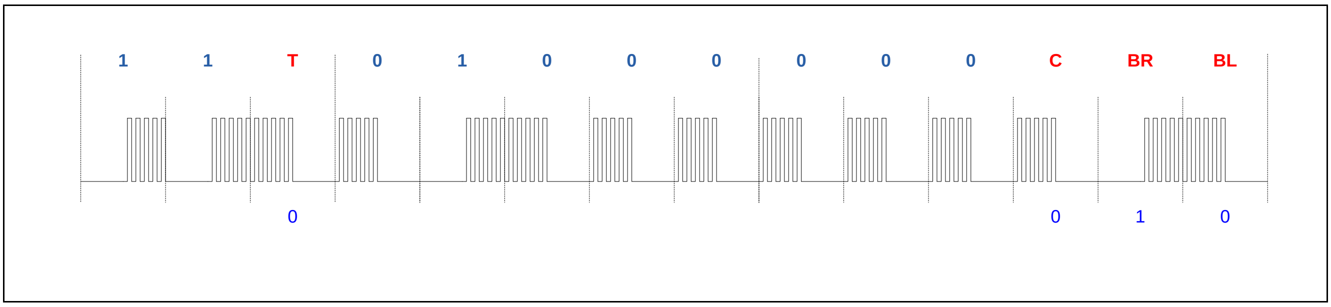


Figure 13 Button-touch packet

The Receiver board also has separate addresses for the wheel. The RC-5 address is **11XXX**. The transmitted angle [**W8:W0**] has 9 bits [**A2:A0,D5:D0**]. The value scales from 0 to 0x11F (0..287 decimal), with 287 being 358.75°, so [**A2:A0**] can be **000**, **001**, **010**, **011** or **100**. In other words, the wheel uses 5 addresses: **11000**, **11001**, **11010**, **11011** and **11100**. The *Receiver* accepts all **11XXX** addresses as wheel angle. The Toggle Bit is changed every time a new angle is transmitted.

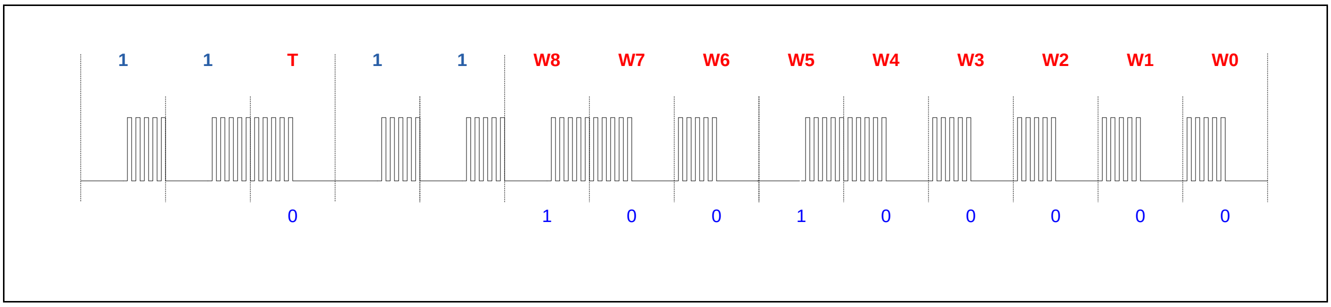


Figure 14 Wheel-angle packet

3.2 Transmission

The *IR Transmitter* transmits the packets on the *COUT61* PWM output. *COUT61* drives the IR diode via a transistor (see Figure 9). The carrier wave is enabled or disabled in two interrupts generated by *CCU6*, the high-performance PWM unit of XC822. *CCU6* contains two flexible 16-bit timers, Timer 12 (*T12*) and Timer 13 (*T13*).

- *T12* generates the carrier wave with a period of 27.8 µs and 25% or 5% duty cycle
- *T13* generates the timing for the transmitted chips with a period of 1.778 ms and a fixed duty cycle of 50% for the RC-5 protocol. *T13* generates an interrupt at the beginning (end) of every bit (*T13PM*) and it generates an interrupt right at the middle of the bit (*T13CM*). In the interrupt service routines (ISR) for these interrupts, the microcontroller enables/disables the carrier wave according to the packet to be transmitted.

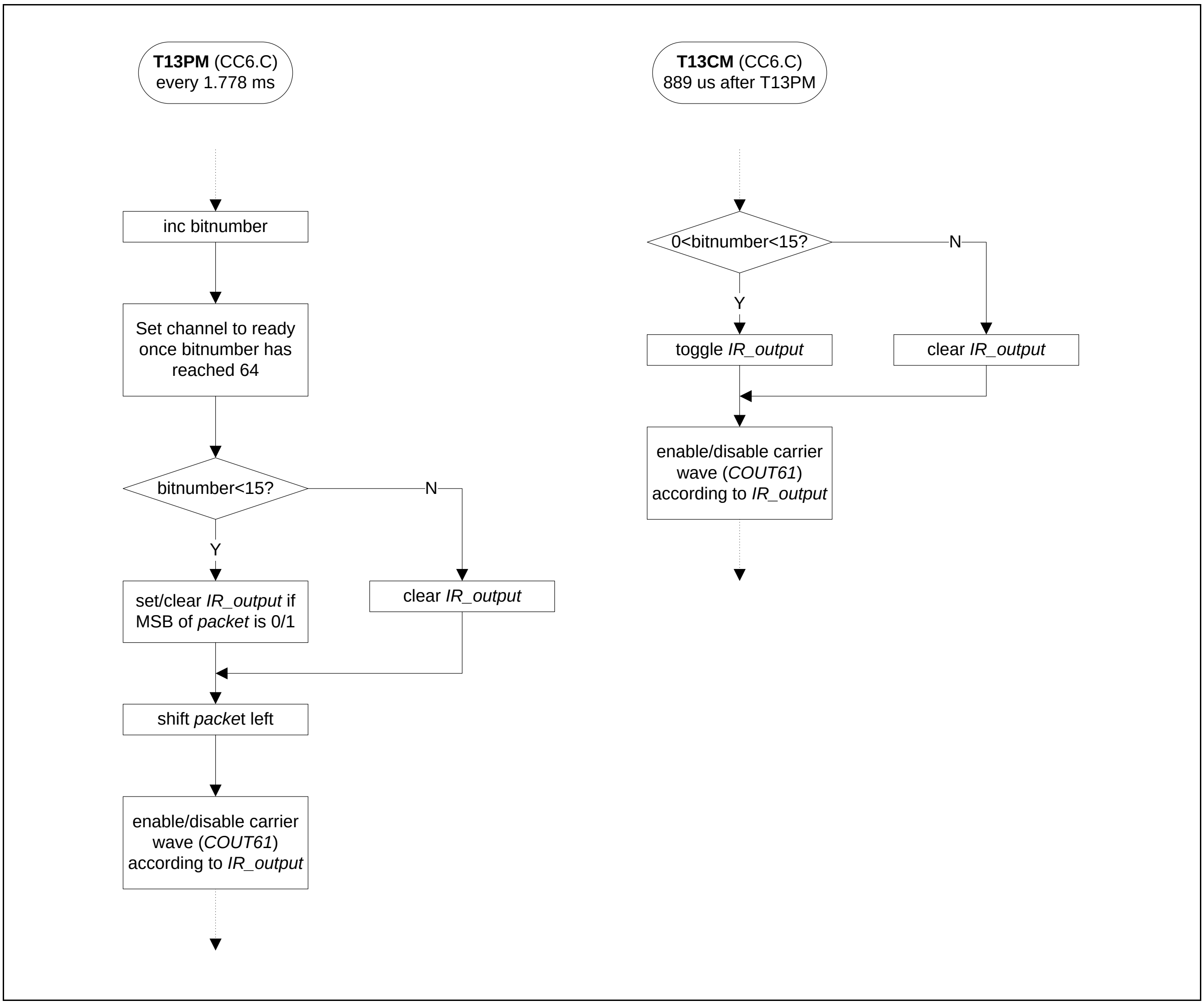

T13PM (CC6.C)
every 1.778 ms
inc bitnumber
Set channel to ready once bitnumber has reached 64
bitnumber<15?
N
Y
set/clear IR_output if MSB of packet is 0/1
clear IR_output
shift packet left
enable/disable carrier wave (COUT61) according to IR_output
T13CM (CC6.C)
889 us after T13PM
0<bitnumber<15?
N
Y
toggle IR_output
clear IR_output
enable/disable carrier wave (COUT61) according to IR_output

Figure 15 IR transmission code sections

3.3 Reception

The receiver IC (*TSOP34836*) on the *IR Receiver* board de-modulates and inverts the received infrared signal so whenever the carrier wave is active, it will output 0, else it will output 1 (see Figure 9). This signal is Manchester-encoded and the microcontroller handles the decoding.

The output of *TSOP34836* is directly connected to the *T2EX* input of the XC822MT microcontroller on the *IR Receiver* board. *T2EX* is the external input of the Timer 2 (*T2*) module of XC822. *T2* is a flexible 16-bit timer with some additional capabilities. Since this module is capable of detecting rising or falling edges on *T2EX*, the microcontroller can detect the rising and falling edges of the demodulated (and inverted) signal and measure the time passed between the edges. In the RC-5 protocol, the distance between edges can be 1.778ms (2 chips) or 0.889 ms (1 chip). The microcontroller measures these distances with a nominal resolution of 0.042 µs. If the measured distance is not 1 or 2 chips, the received edge is deemed noise and the microcontroller will wait for a new packet. A tolerance of about 15% is allowed to be robust against ambient noise and to accommodate internal oscillator imperfections.

Detecting the edges and knowing the distance between them is enough to decode a Manchester-encoded signal. The first bit is always 1 so the first edge is always a falling edge. Every falling edge is followed by a rising edge and every rising edge is followed by a falling edge. Whenever an edge is detected, *T2* generates a capture event interrupt where the received signal is evaluated. If no edge is detected, *T2* overflows and re-initializes every 2.73 ms.

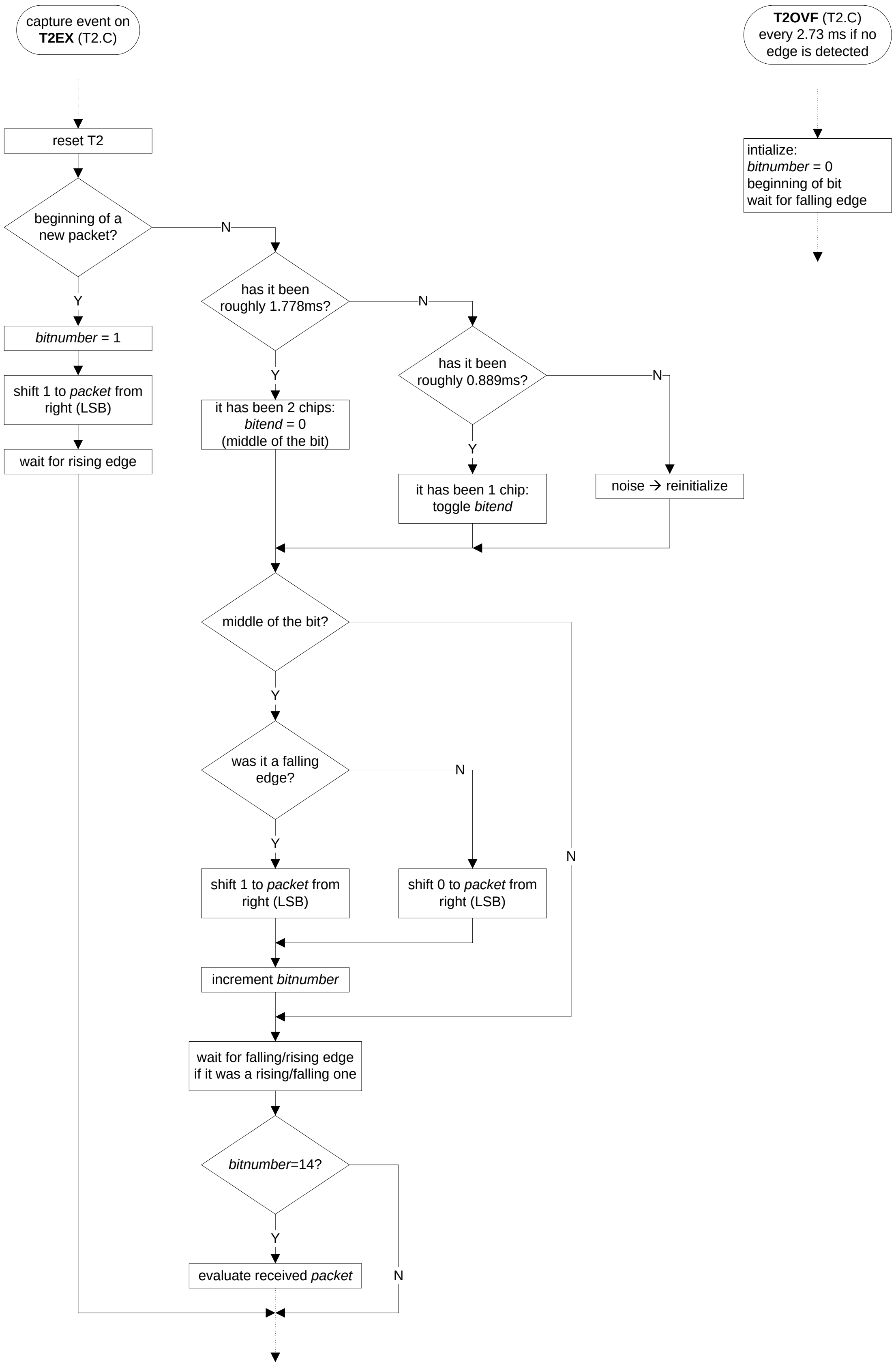

capture event on T2EX (T2.C)
reset T2
beginning of a new packet?
N
Y
bitnumber = 1
shift 1 to packet from right (LSB)
wait for rising edge
has it been roughly 1.778ms?
N
Y
it has been 2 chips: bitend = 0 (middle of the bit)
has it been roughly 0.889ms?
N
Y
it has been 1 chip: toggle bitend
noise → reinitialize
middle of the bit?
Y
N
was it a falling edge?
N
Y
shift 1 to packet from right (LSB)
shift 0 to packet from right (LSB)
increment bitnumber
wait for falling/rising edge if it was a rising/falling one
bitnumber=14?
Y
N
evaluate received packet
T2OVF (T2.C) every 2.73 ms if no edge is detected
intialize:
bitnumber = 0
beginning of bit
wait for falling edge

Figure 16 IR reception code sections

Once the RC-5 packet is received, the microcontroller determines whether it contains new button information, calibration information or a new angle. On the *USB Docking Station*, *LED7* shows the status of the left button (turned on if touched and turned off if untouched) and *LED5* shows the status of the right button.

By default the 6 other LEDs show the angle of the wheel. The brightness of the LEDs is controlled by the duty cycle of a 500Hz PWM signal generated by T12 in the CCU6 module. At any given moment a maximum of 2 LEDs are on, with dimmed brightness to indicate the position of the finger. For example, at 45° LED4 and LED3 are on; LED4 has 75% brightness and LED3 has 25% brightness. The LEDs are enabled and disabled by CCU6 multi-channel mode (MCM). The LEDs are updated every 5 ms in the *T13PM* interrupt.

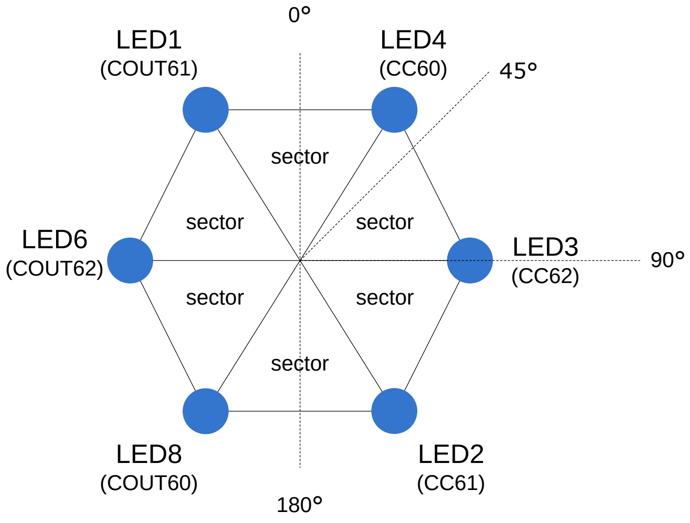


Figure 17 Touch Wheel LEDs on the USB Docking Station

Alternative modes:

- If the user connects to the *USB Docking Station*, the behavior of the LEDs can be changed from the PC (via *U-SPY*, see chapter 6) so that the all LEDs are turned on and their brightness is adjusted by dialing.
- During calibration, the 6 LEDs blink in an alternating triangle pattern. This overrides the function in which they indicate angle.

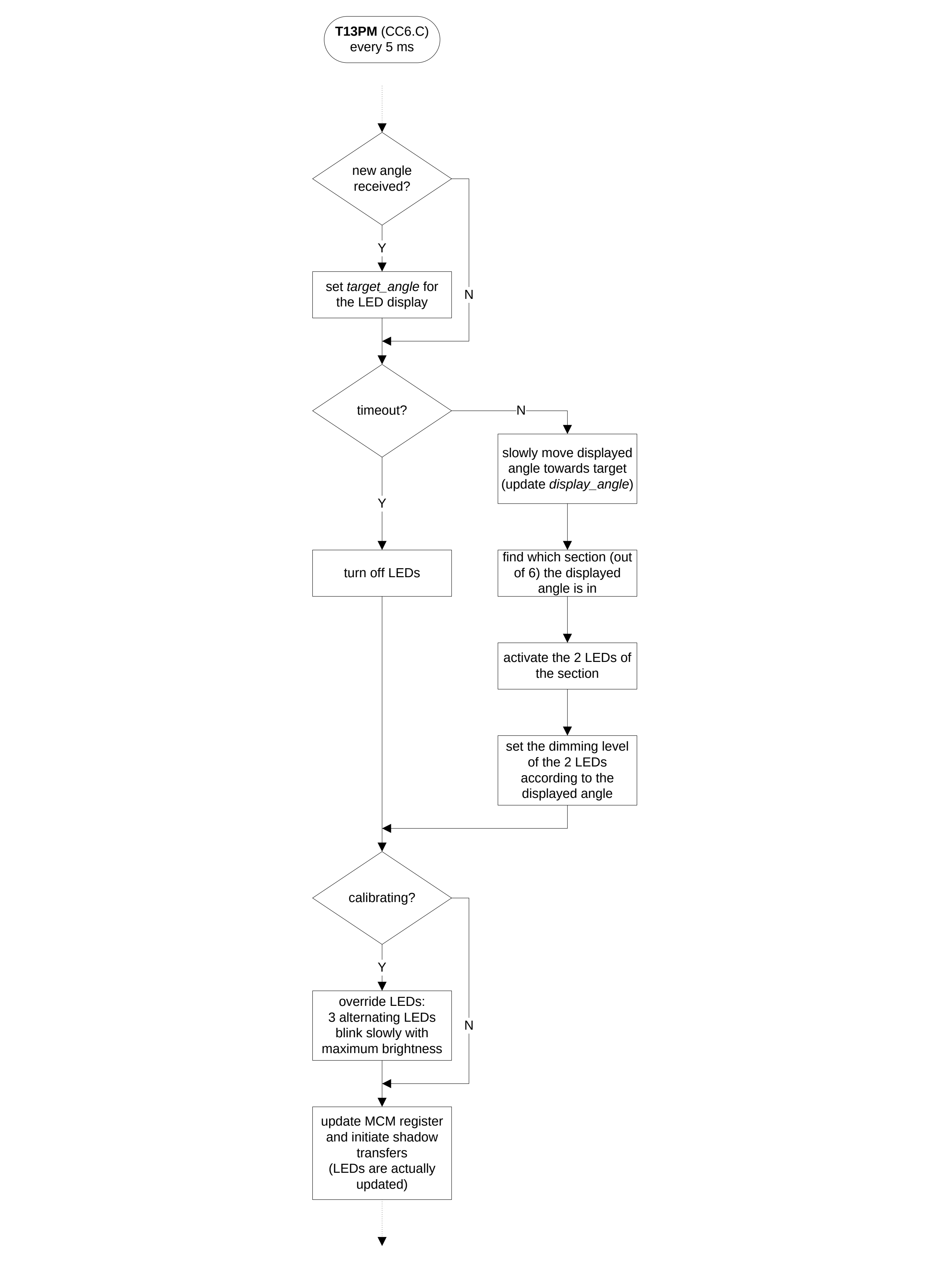

T13PM (CC6.C)
every 5 ms
new angle received?
Y
N
set *target_angle* for the LED display
timeout?
N
Y
slowly move displayed angle towards target (update *display_angle*)
turn off LEDs
find which section (out of 6) the displayed angle is in
activate the 2 LEDs of the section
set the dimming level of the 2 LEDs according to the displayed angle
calibrating?
Y
N
override LEDs: 3 alternating LEDs blink slowly with maximum brightness
update MCM register and initiate shadow transfers (LEDs are actually updated)

Figure 18 Angle display on the wheel LEDs

4 Touch Interface

The *IR Transmitter* has five touch pads; two are used as touch buttons and the remaining three form a wheel for dialing. All 5 pads are handled by the XC822MT microcontroller's LED Touch-Sense Control Unit (*LEDTSCU*) which is a dedicated touch-sense controller module.

The main touch sensing functions, handled by software, are:

- Sample accumulation (ROM library)
- Signal filtering and moving average generation (ROM library)
- Touch detection (ROM library)
- Touch wheel calibration (user software in Flash)
- Signal tuning (user software in Flash)

If properly configured, the LEDTSCU automatically measures the capacitance of the five pads. This capacitance increases when a button is touched. A library function in ROM processes the capacitance signals and detects touches on the two buttons. It does so by accumulating 3 samples and low-pass filtering them to create a moving average. The moving average filters noise and is used as a comparison platform to detect sudden changes in capacitance. When a button is touched or released, a corresponding pad flag in RAM (refer to the User's Manual) will be set or reset and the *IR Transmitter* will transmit a button-press or release command to the *IR Receiver* which will turn on or off the corresponding LED on the *USB Docking Station*.

The pad flags for the wheel pads are unused (always cleared) and it is the moving averages ("*pad averages*") that are used instead to calculate the angle of the touch. The three pads are automatically calibrated to the same sensitivity and resolution during startup. The parameters are optimized for 3V input voltage (battery operated) but some adjustments are automatically made if 5V input voltage is detected (docked in the *USB Docking Station*). Once the pad averages are stable, an angle calculation algorithm is run if the wheel is touched. If a new angle is calculated, the *IR Transmitter* will start to transmit a new-angle command to the *IR Receiver* which will show the angle on the 6 touch wheel LEDs on the *USB Docking Station*.

4.1 Wheel Angle Calculation

The three touch pads of the wheel are placed in a spatially interpolated manner.

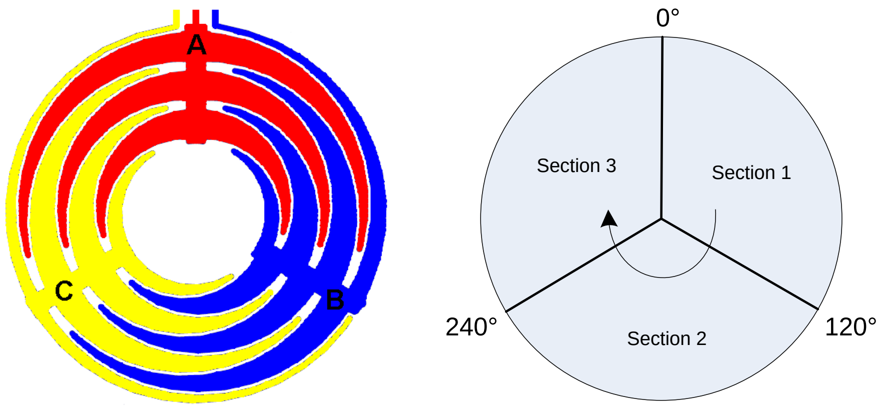


Figure 19 Spatially interpolated wheel layout and abstraction

If the pads are calibrated to roughly the same sensitivity and the wheel is dialed clockwise with constant angular speed and constant pressure (constant effective finger area), the pad average signals are expected to behave in a linear manner in this model as seen in Figure 20.

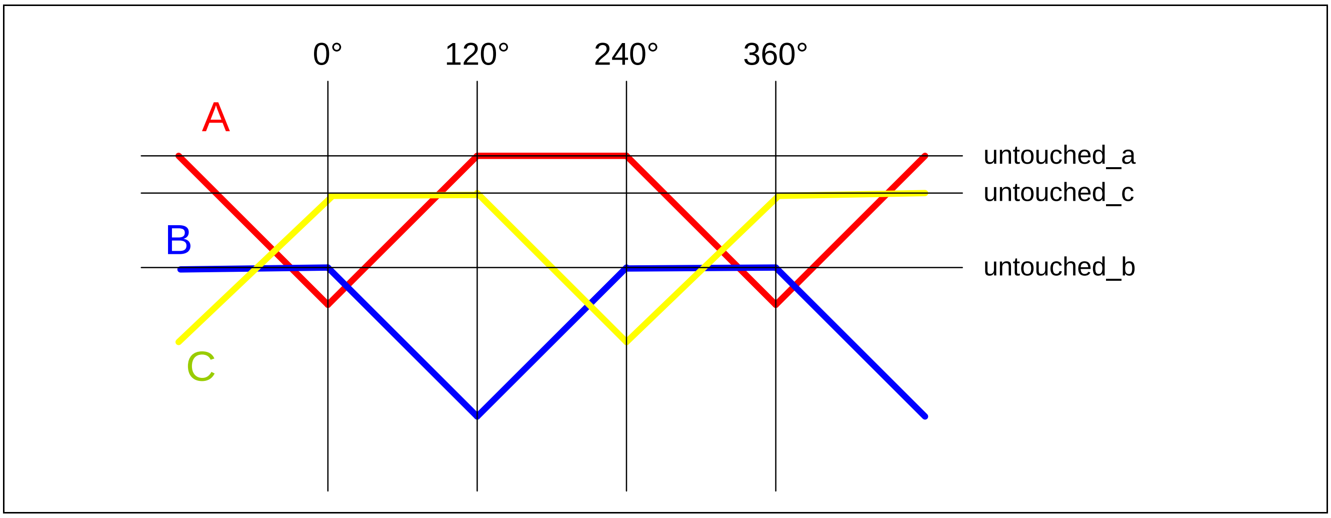


Figure 20 Pad average signals of the three wheel pads during dialing

Values *untouched_a*, *untouched_b* and *untouched_c* are the pad average levels for pad A, B and C respectively when they are not touched.

If the pads have roughly the same sensitivity, the three signals can be tuned to have a common untouched level.

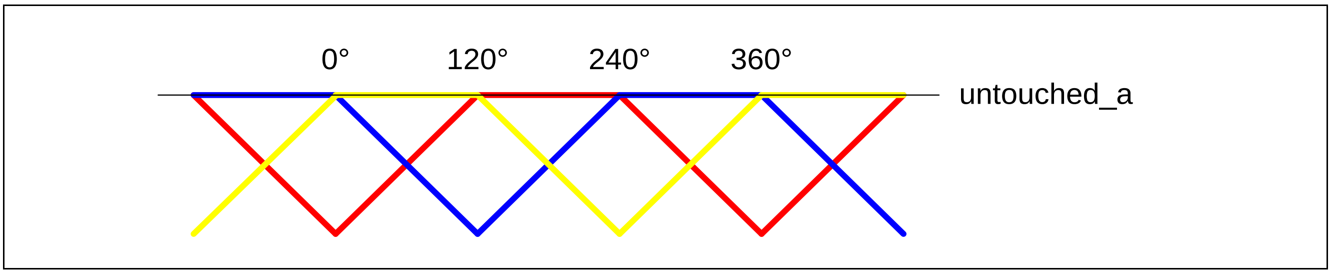


Figure 21 Pad average signals of the three wheel pads after tuning

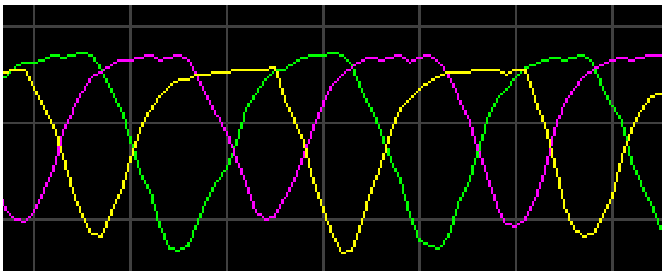


Figure 22 Actual pad average signals after tuning

The, now common, untouched level is very high compared to the difference between touched and untouched states. To make calculations easier, the signals are transformed near to zero by linear combinations. This also makes the transitions between angle sections smooth, which is especially important if the three pads have different sensitivity or unstable untouched levels due to imperfect calibration or a changing environment.

$$X = \frac{A+B}{2} - C \qquad Y = \frac{A+C}{2} - B \qquad Z = \frac{B+C}{2} - A$$

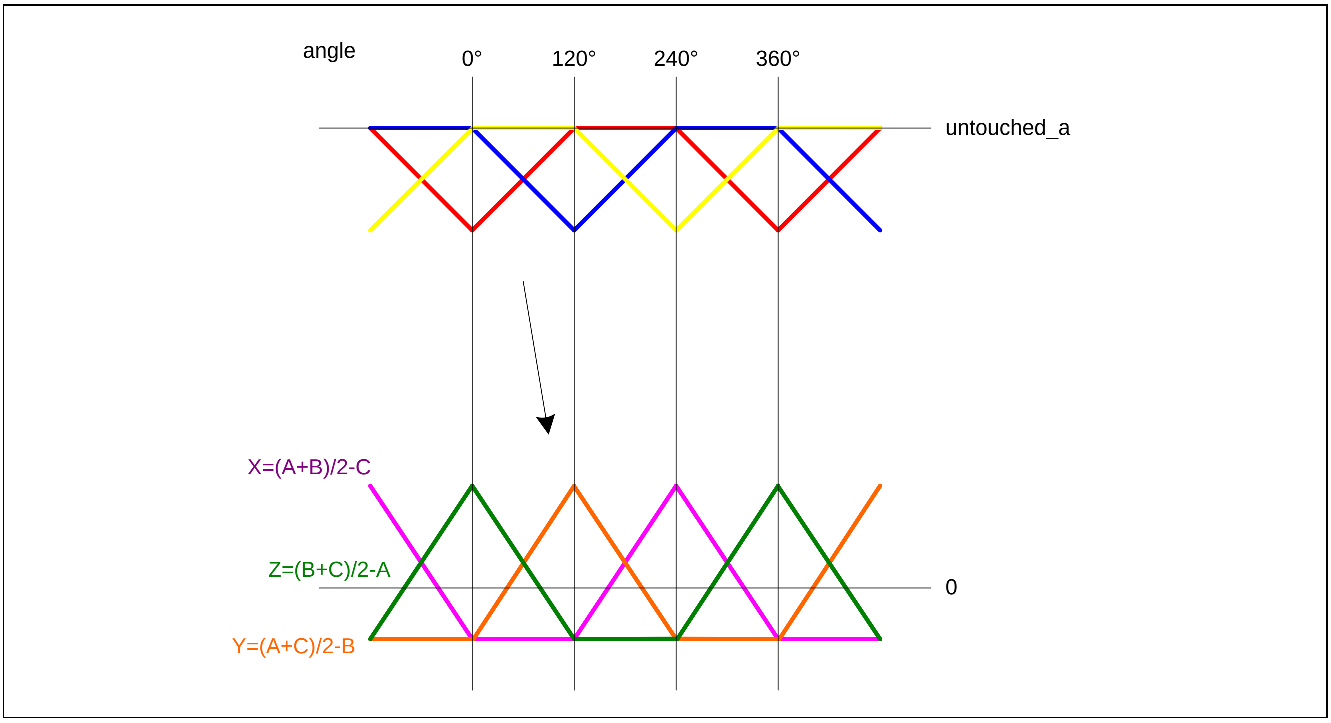


Figure 23 Combined pad average signals

The resulting *X*, *Y* and *Z* signals still have three distinct sections between 0° and 360°.

Section 1 (0° to 120°)

Section 1 before the transformation has three signals between *UT* and *UT-MAXT*. *UT* stands for the untouched level and *UT-MAXT* stands for the signal level when the largest area of the respective pad is touched (this happens at 0°, 120° and 240°).

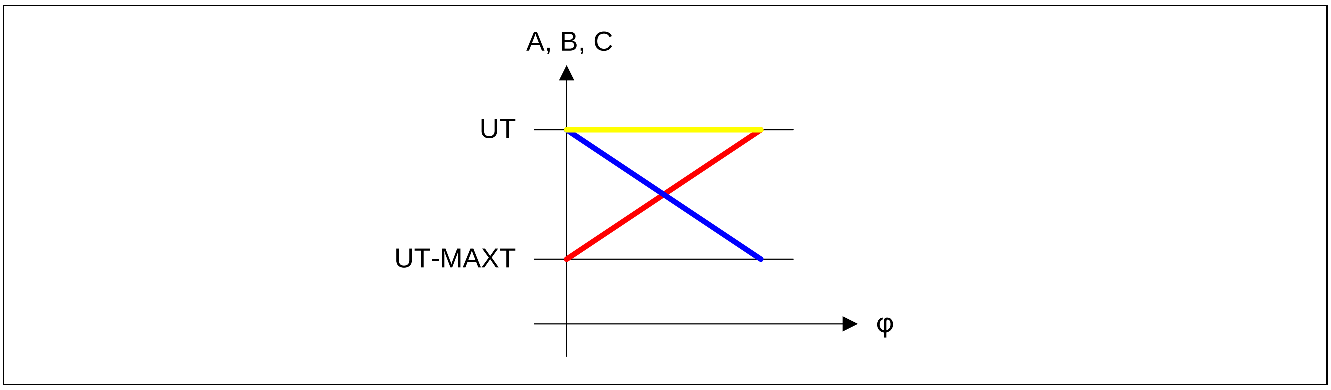


Figure 24 Section 1 before transformation

After the transformation, the *X*, *Y*, *Z* signals have much lower values. The angle axis has been arbitrarily scaled from -1 to 2 in this region for convenience.

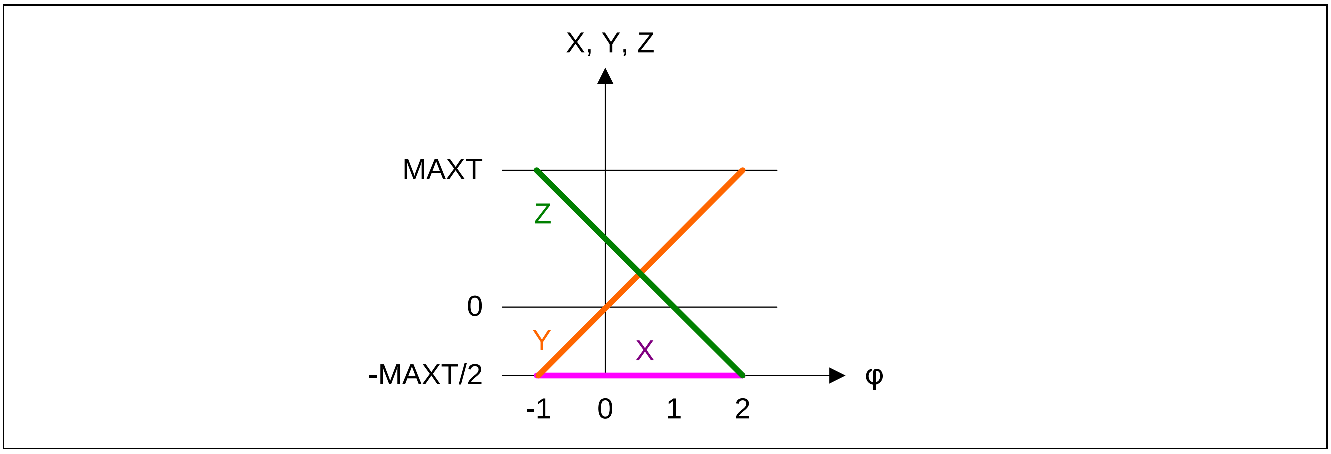


Figure 25 Section 1 after transformation

Signal *X* (see Figure 25) is constant low in this section so it does not participate in the angle calculation. The other two signals can be described as:

I.

$$Y = \frac{MAXT}{2}\varphi$$

II.

$$Z = \frac{MAXT}{2} - \frac{MAXT}{2}\varphi$$

If we rearrange **I**, we get $\frac{MAXT}{2} = \frac{Y}{\varphi}$ which we can substitute in **II**:

$$Z = \frac{Y}{\varphi} - \frac{Y}{\varphi}\varphi$$

$$Z = \frac{Y}{\varphi}(1-\varphi)$$

$$\varphi(Y+Z) = Y$$

III.

$$\varphi = \frac{Y}{Y+Z}$$

IV.

$$\varphi = 1 - \frac{Z}{Y+Z}$$

One division is needed to calculate the angle; this operation needs the most computing performance. To minimize error, it is safer to use **III** if *Y* is larger and **IV** if *Z* is larger.

An offset of 1 and a scaling factor of 2^*R* are added to create a more usable calculated angle. *R* is for *resolution* and corresponds to the number of left bitshifts on the numerator.

Section 1 Left

$$\varphi = 2 \times 2^R - \frac{Z \times 2^R}{Y+Z}$$

Section 1 Right

$$\varphi = \frac{Y \times 2^R}{Y+Z} + 2^R$$

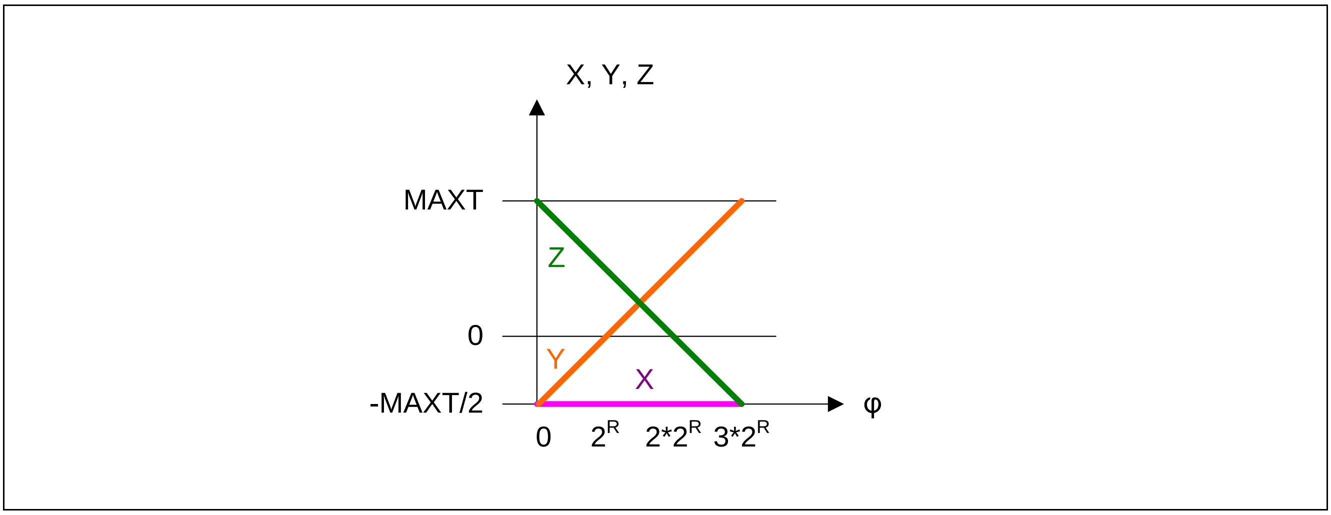


Figure 26 Section 1 after offsetting and scaling

Sections 2 and 3

In these two sections, the angle can be calculated in a similar way as in section 1, using the two non-constant signals. Offsets of 4 and 7, and the same scaling factor, can then be added to sections120°-240° and 240°-360° respectively to get a calculated angle of 0..9*2R for 0°..360°.

Section 2 Left

$$\varphi = 5 \times 2^R - \frac{Y \times 2^R}{X + Y}$$

Section 2 Right

$$\varphi = \frac{X \times 2^R}{X + Y} + 4 \times 2^R$$

Section 3 Left

$$\varphi = 8 \times 2^R - \frac{X \times 2^R}{X + Z}$$

Section 3 Right

$$\varphi = \frac{Z \times 2^R}{X + Z} + 7 \times 2^R$$

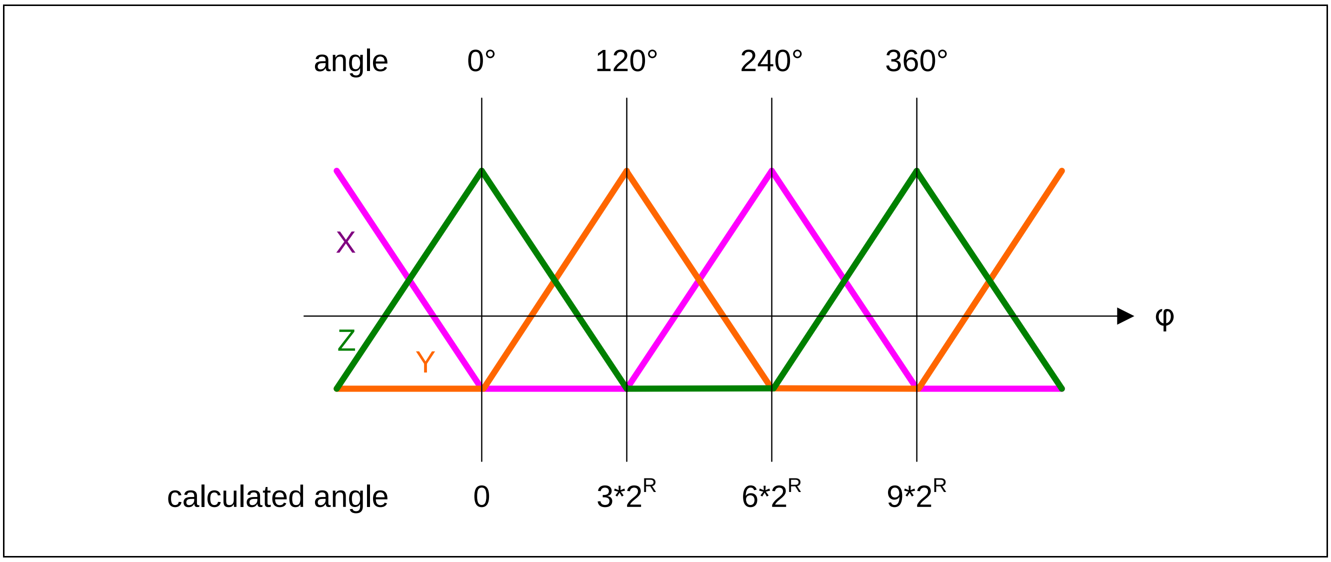


Figure 27 Calculated angle vs real angle across all sections

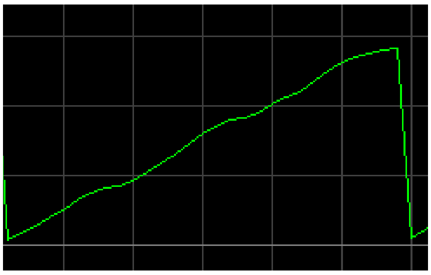


Figure 28 Actual calculated angle for a full round

Infineon provides a function library for angle calculation. The *resolution*, explained earlier, is user selectable from 1 to 8. The XC822M and XC822MT microcontrollers have a Multiplication/Division Unit (*MDU*) for hardware acceleration. If the *MDU* is used for the division necessary to calculate the angle, the *resolution* is fixed at 8, execution is faster and the code size is about 250 bytes smaller than without hardware acceleration. The disadvantage is that the MDU increases the microcontroller’s current consumption by almost 1 mA.

5 Power Saving

It is essential for battery-operated devices to minimize power consumption. To achieve low average current consumption the microcontroller of the IR Transmitter runs at a reduced clock speed, uses a low voltage power supply, and dynamically changes operating mode dependent on usage.

Static measures

- The microcontroller runs on an 8 MHz system clock instead of 24 MHz to reduce power consumption in active mode.
- All unused peripherals are permanently disabled by gating off their clock inputs. Only *LEDTSCU*, *T2* and *CCU6* are enabled.

Dynamic measures

- When no interrupt service routine is running, the microcontroller is in idle mode. In idle mode the CPU clock is stopped.
- If the IR Transmitter is unused for more than 10 s, the microcontroller enters *power down mode 2* and wakes up every 300 ms for a short time to check for user activity. In *power down mode 2*, the main embedded voltage regulator is switched off; only the low-power embedded voltage regulator keeps operating. The flash memory and the main oscillator are put in power-down mode too. The onboard 75 kHz oscillator and the Real Time Clock module remain active and generate a periodic wakeup signal every 300 ms. If there is no user activity on the touch buttons or the touch wheel, the microcontroller goes back to power down mode in 1.8 ms after waking up.
- CCU6 is turned off before the microcontroller enters power-down mode to reduce power consumption during the periodic wakeups. It is turned back on if the wheel or the buttons have been touched.

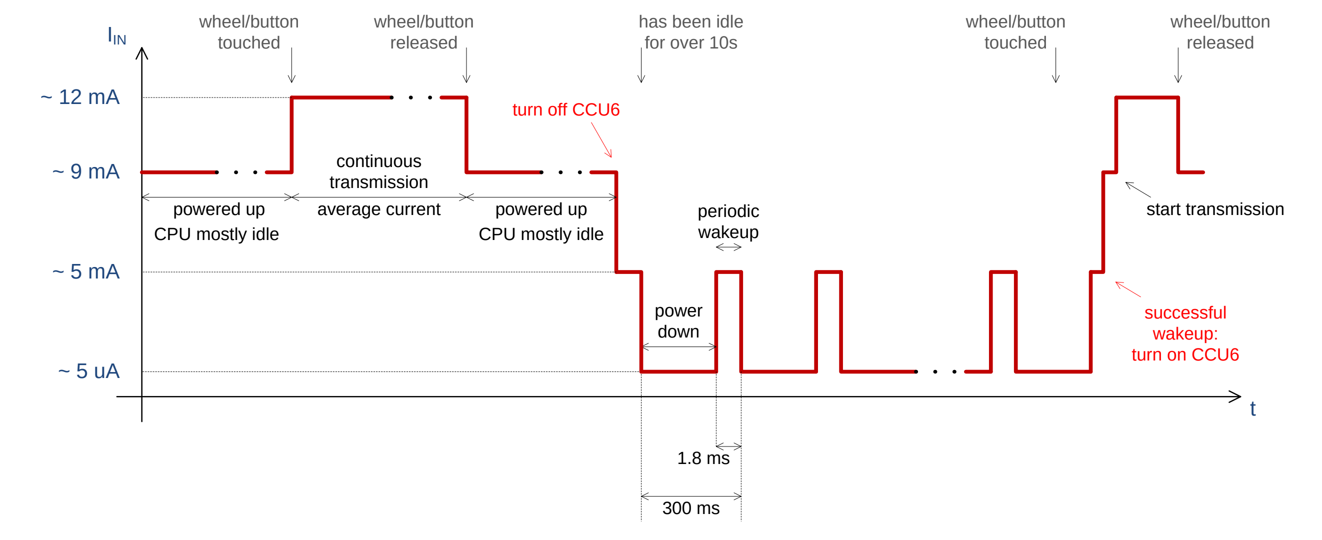
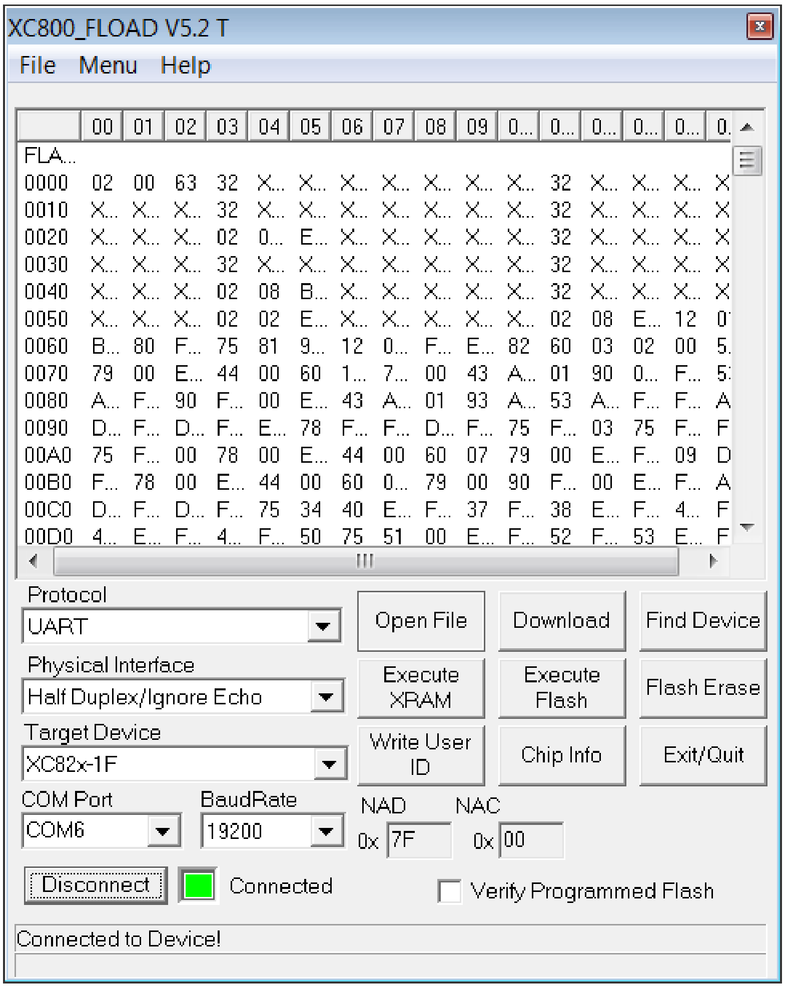


Figure 29 Average current consumption in different operating modes

6 Programming Access

The *USB Docking Station* provides programming access to the microcontrollers on both the *IR Transmitter* and *IR Receiver* boards, when the onboard switch is set to the 'L' position. The USB Docking Station contains an FTDI chip, FT232RL, which acts as a USB-to-UART bridge and also controls the microcontoller's emulated MBC (P2.1) and RESET (P2.2) pins. Programming access is wired for half-duplex UART on pin P0.6. Flash content can be modified with the XC800 FLOAD tool which is integrated into DAVE™ BENCH and is also available in a stand-alone version.



	00	01	02	03	04	05	06	07	08	09	0...	0...	0...	0...	0...	0.
FLA...																
0000	02	00	63	32	X...	X...	X...	X...	X...	X...	X...	32	X...	X...	X...	X
0010	X...	X...	X...	32	X...	X...	X...	X...	X...	X...	X...	32	X...	X...	X...	X
0020	X...	X...	X...	02	0...	E...	X...	X...	X...	X...	X...	32	X...	X...	X...	X
0030	X...	X...	X...	32	X...	X...	X...	X...	X...	X...	X...	32	X...	X...	X...	X
0040	X...	X...	X...	02	08	B...	X...	X...	X...	X...	X...	32	X...	X...	X...	X
0050	X...	X...	X...	02	02	E...	X...	X...	X...	X...	X...	02	08	E...	12	0
0060	B...	80	F...	75	81	9...	12	0...	F...	E...	82	60	03	02	00	5.
0070	79	00	E...	44	00	60	1...	7...	00	43	A...	01	90	0...	F...	5:
0080	A...	F...	90	F...	00	E...	43	A...	01	93	A...	53	A...	F...	F...	A
0090	D...	F...	D...	F...	E...	78	F...	F...	D...	F...	75	F...	03	75	F...	F
00A0	75	F...	00	78	00	E...	44	00	60	07	79	00	E...	F...	09	D
00B0	F...	78	00	E...	44	00	60	0...	79	00	90	F...	00	E...	F...	A
00C0	D...	F...	D...	F...	75	34	40	E...	F...	37	F...	38	E...	F...	4...	F
00D0	4...	E...	F...	4...	F...	50	75	51	00	E...	F...	52	F...	53	E...	F

Protocol: UART
Physical Interface: Half Duplex/Ignore Echo
Target Device: XC82x-1F
COM Port: COM6
BaudRate: 19200
NAD: 0x 7F
NAC: 0x 00
🟩 Connected
□ Verify Programmed Flash
Connected to Device!

Figure 30 XC800 Fload

FLOAD can be used to access whichever board is plugged in to the USB Docking Station.

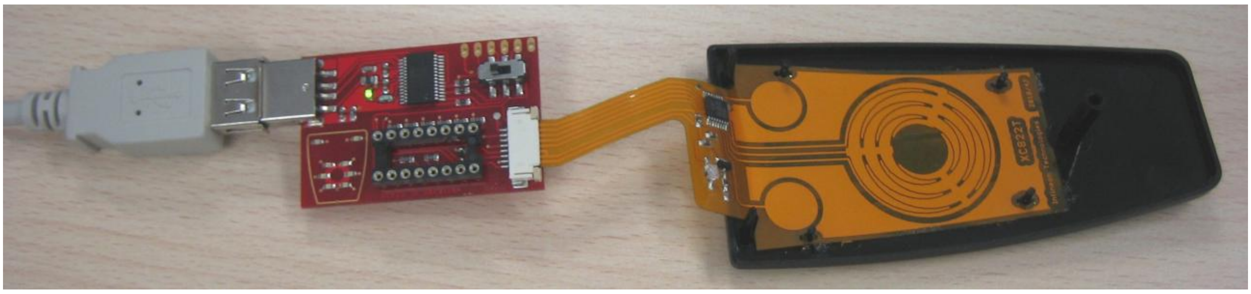


Figure 31 IR Transmitter connected to USB Docking Station

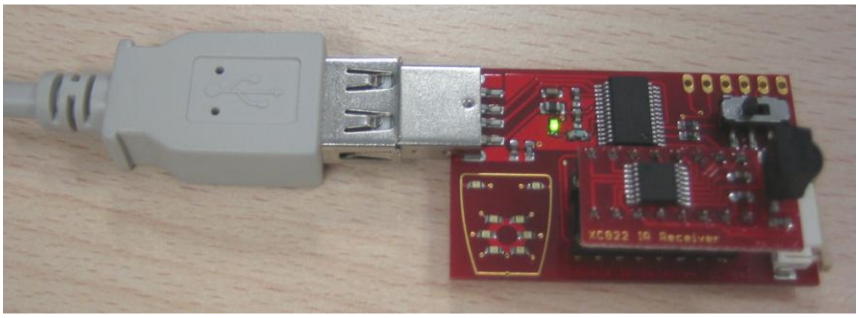


Figure 32 IR Receiver connected to USB Docking Station

If both the *IR Transmitter* and the *IR Receiver* are plugged in, then the *IR Transmitter* takes priority and it is the only one that can be accessed.

The XC822 boot configuration does not depend on any pin status during reset. Instead, a Boot Mode Index (BMI) configuration determines the entry to various boot modes such as User Mode, Boot-Loader (BSL) Mode and On-chip Debug (OCDS) Mode. After reset, the BMI value is read and the respective boot mode entry is automatically executed.

The microcontrollers on the *IR Transmitter* and *IR Receiver* boards are both programmed to "User Mode (Productive)". In this mode, the Boot ROM jumps to the program memory address 0x0000 on startup to execute the user code in the Flash memory. This mode provides Flash memory protection from external access (read/write).

Changing the BMI value to enter another boot mode is achieved by programming a specific code embedded in the user code. It is located in MAIN.C at the *(MAIN_main,2)* section. This section only gets executed once after RESET. This code checks GPIO pin P2.1 (MBC emulation) and if it is low, a Boot ROM routine is called to re-enter UART Boot-Loader Mode. Once UART Boot-Loader Mode is entered, the user can change the contents of the Flash memory. This specific user code must be present in Flash and executed under certain conditions to ensure that programming the microcontroller is possible.

RESET happens during power-on and is also emulated on pin P2.2. Pulling down P2.2 momentarily generates an interrupt (External Interrupt 2) and the interrupt service routine generates a software RESET. The FTDI chip on board the *USB Docking Station* automatically controls these pins when the user tries to access the microcontrollers with FLOAD*.*

7 Monitoring

7.1 U-SPY

U-SPY is a UART terminal program which allows the user to view a serial communication through a PC serial port. Its features include transmission of a byte or group of bytes, configuration of protocol for bytes transmission/reception and creation of dedicated control buttons, display fields, progress bars and oscilloscope for better visualization.

For more information on U-SPY, please refer to the Help menu in U-SPY.

The U-SPY can be launched directly from DAVE^{TM} Bench by clicking on the ⇄ icon.

7.2 Settings

The custom configuration and user interface for a particular task or application can be saved in the format "*xxx.ini*". This allows specific setting files to be shared among users.

For the IR Remote Controller, two *.ini* files have been configured, **RemoteControl_Rotation.ini** and **RemoteControl_TouchSense&WheelEvaluation.ini**. Serial communication is via the **full-duplex UART** protocol at a baudrate of **57.6 kbps**.

*Note: Ensure that the USB docking station mode is switched to "**R**" (**Figure 33**) before running any of the monitoring routines.*

In "R" mode ("RUN"), the UART interface will be full-duplex, whereas in "L" mode ("LOAD"), the UART interface will be half-duplex.

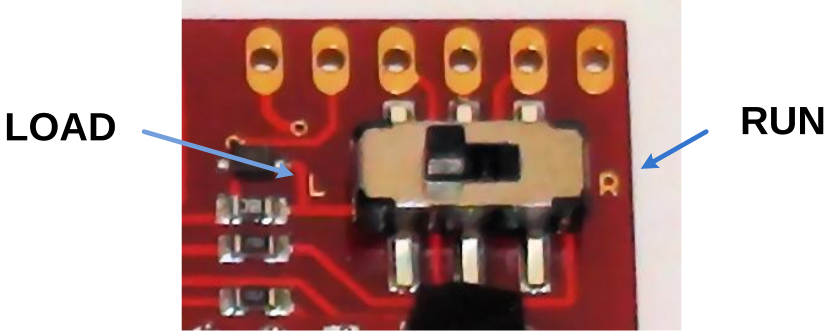


Figure 33 Switch to "RUN" mode on USB Docking Station

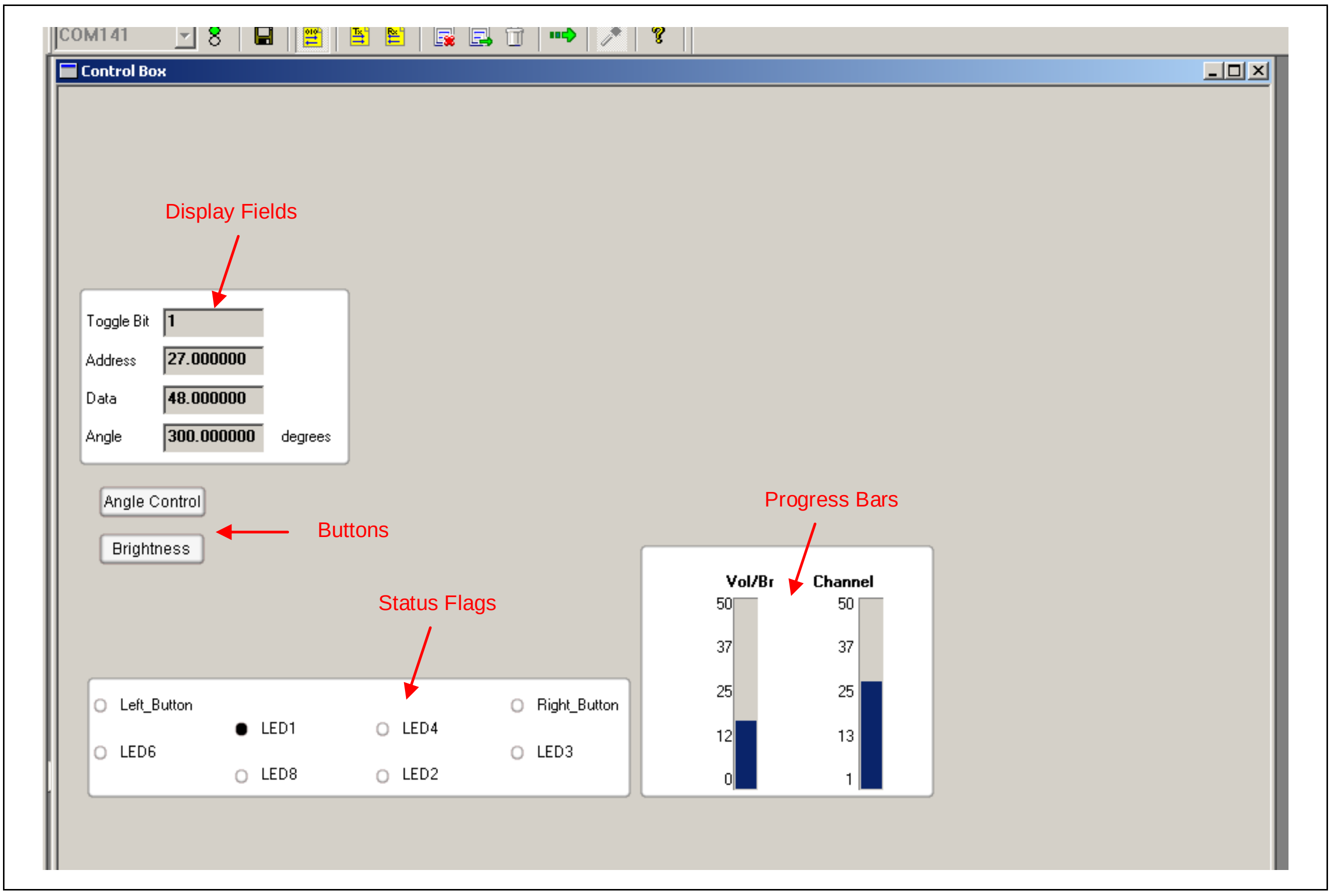
7.3 UART Interrupt

Any data transmission to or from U-SPY will trigger the UART interrupt in the docked microcontroller. Checks are performed during the interrupt to determine whether data is to be transmitted or received. The data transmit or receive process is then carried out automatically.

7.4 RemoteControl_Rotation.ini

This settings file (**Figure 34**) is customized to allow the user to monitor the 14-bits RC-5 packet decoded by the receiving microcontroller as well as an interface between the IR Remote Controller hardware and software.

Note: This settings file is only to be used with the IR Receiver board plugged in to the USB docking station.



Display Fields

Toggle Bit: 1
Address: 27.000000
Data: 48.000000
Angle: 300.000000 degrees

Buttons

Status Flags

Progress Bars

Figure 34 RemoteControl_Rotation.ini User Interface

7.4.1 Data Format

A data group consisting of 8 bytes is transmitted to U-SPY, in the following format (**Table 1**):

Table 1 RemoteControl_Rotation.ini Transmitted Data Format

	DataByte0(D0)	D1	D2	D3	D4	D5	D6	D7
Value (hex)	83	XX	XX	XX	XX	XX	XX	XX
Description	I.D.	Angle	Channel	Volume/ Brightness	LEDs	Toggle Bit	Address	Data

The data group received by U-SPY is then matched or masked, before it is displayed.

Display Fields

The received RC-5 packet bits (Toggle bit, Address and Data) as described in **Chapter 3.1: Protocol** and the calculated Wheel Angle are displayed in the display fields.

Buttons

The buttons are used to transmit data from U-SPY to the receiver board. Data is transmitted in the following format (**Table 2**):

Table 2 U-SPY to Microcontroller Transmitted Data Format

	D0	D1
Value (hex)	08	XX
Description	I.D.	Button no.

Based on this data received by the receiver board, the display mode can be switched between Angle and Brightness Control via the toggling of a bit data in the code.

Status Flags

The statuses of the LEDs received by U-SPY are masked before they are displayed as status flags. It is important that the bits of a mask do not overlap with the bits of another mask. This is to ensure that status flags are not falsely turned on. The masks used are as follows (**Table 3**):

Table 3 LED masks for Status Flags

LED Number	1	2	3	4	5 (Right Button)	6	7 (Left Button)	8
Mask (hex)	80	10	08	04	02	40	01	20

Progress Bars

Channel and Volume/Brightness are displayed as progress bars. Channel increases with the press of the right button and decreases with the press of the left button. Volume/Brightness increases with the clockwise dialing of the wheel and decreases with the anti-clockwise dialing of the wheel.

7.5 RemoteControl_TouchSense&WheelEvaluation.ini

This settings file (**Figure 35**) is customized to allow the user to monitor parameters of the Touch Wheel Library and the LEDTS ROM Library.

Note: This settings file is to be used with the transmitter board plugged in to the USB docking station.

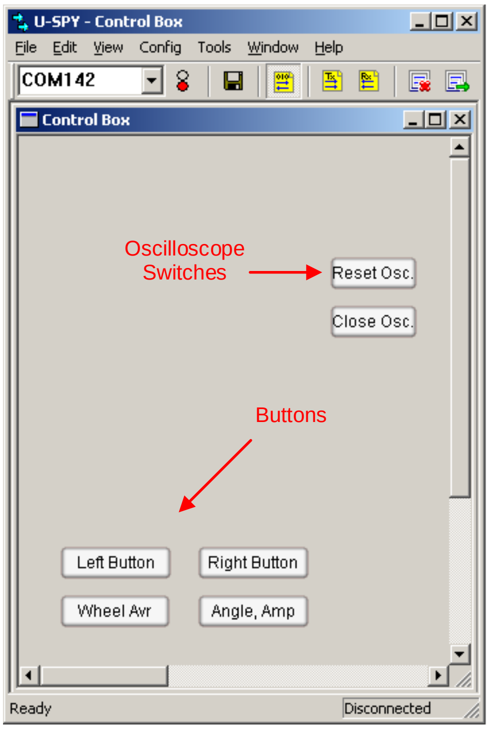


Figure 35 RemoteControl_TouchSense&WheelEvaluation.ini User Interface

7.5.1 Data Format

In this settings file, the data group transmitted to U-SPY is solely for the display of signals on the oscilloscope. The format of the data group, also consisting of 8 data bytes, is as follows (**Table 4**):

Table 4 RemoteControl_TouchSense&WheelEvaluation.ini Transmitted Data Format

	DataByte0(D0)	D1	D2	D3	D4	D5	D6	D7
Value (hex)	B3	B3	XX	XX	XX	XX	XX	XX
Description	I.D.	I.D.	Signal1 High Byte	Signal1 Low Byte	Signal2 High Byte	Signal2 Low Byte	Signal3 High Byte	Signal3 Low Byte

The data group received by U-SPY is then matched before it is displayed.

Buttons

In this settings file, the buttons allow the user to choose the signal which the user would like to monitor. The description of the data format for the buttons is the same as in the previous settings file. Data is transmitted in the following format (**Table 5**):

Table 5 U-SPY to Microcontroller Transmitted Data Format

	D0	D1
Value (hex)	08	XX
Description	I.D.	Button no.

The data received by the microcontroller will be used to determine the signals that will be transmitted to U-SPY for display on the oscilloscope.

Oscilloscope

The oscilloscope function allows the user to monitor up to 3 signals at a time.

As mentioned in the previous section, the user is able to monitor four different types of signals in this settings file. The signals displayed in each mode are as follows (**Table6: Left/Right Button**, **Table7: Wheel Avr**, **Table8: Angle, Amp**):

Table 6 Signals Displayed for Left/Right Button

	Signal1	Signal2	Signal3
Description	Pad Total_TSCTR * 2^{DIVISORN}	Pad Average	None
Colour	Green	Pink	Yellow

Table 7 Signals Displayed for Wheel Avr

	Signal1	Signal2	Signal3
Description	Wheel Pad1 Average	Wheel Pad2 Average	Wheel Pad3 Average
Colour	Green	Pink	Yellow

Table 8 Signals Displayed for Angle, Amp

	Signal1	Signal2	Signal3
Description	Calculated Wheel Angle	Wheel Amplitude	None
Colour	Green	Pink	Yellow

8 Schematics and Layout

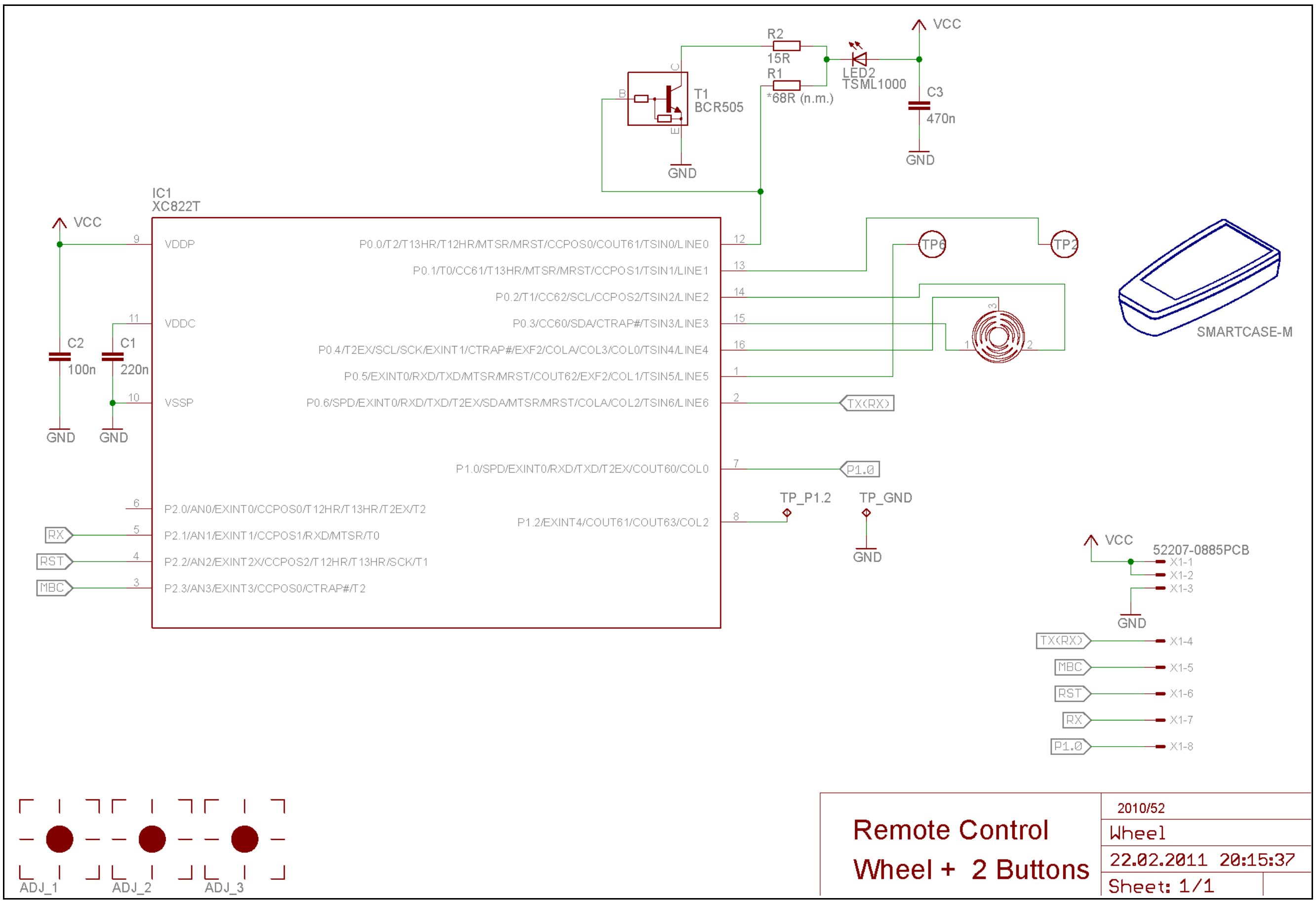

R2
15R
R1
*68R (n.m.)
LED2
TSML1000
VCC
C3
470n
GND
T1
BCR505
B
C
E
GND
IC1
XC822T
VCC
9
VDDP
11
VDDC
10
VSSP
C2
100n
C1
220n
GND
GND
P0.0/T2/T13HR/T12HR/MTSR/MRST/CCPOS0/COUT61/TSIN0/LINE0
12
P0.1/T0/CC61/T13HR/MTSR/MRST/CCPOS1/TSIN1/LINE1
13
P0.2/T1/CC62/SCL/CCPOS2/TSIN2/LINE2
14
P0.3/CC60/SDA/CTRAP#/TSIN3/LINE3
15
P0.4/T2EX/SCL/SCK/EXINT1/CTRAP#/EXF2/COLA/COL3/COL0/TSIN4/LINE4
16
P0.5/EXINT0/RXD/TXD/MTSR/MRST/COUT62/EXF2/COL1/TSIN5/LINE5
1
P0.6/SPD/EXINT0/RXD/TXD/T2EX/SDA/MTSR/MRST/COLA/COL2/TSIN6/LINE6
2
TX(RX)
P1.0/SPD/EXINT0/RXD/TXD/T2EX/COUT60/COL0
7
P1.0
P1.2/EXINT4/COUT61/COUT63/COL2
8
TP_P1.2
TP_GND
GND
6
P2.0/AN0/EXINT0/CCPOS0/T12HR/T13HR/T2EX/T2
RX
5
P2.1/AN1/EXINT1/CCPOS1/RXD/MTSR/T0
RST
4
P2.2/AN2/EXINT2X/CCPOS2/T12HR/T13HR/SCK/T1
MBC
3
P2.3/AN3/EXINT3/CCPOS0/CTRAP#/T2
TP6
TP2
1
2
3
SMARTCASE-M
VCC
52207-0885PCB
X1-1
X1-2
X1-3
GND
TX(RX)
X1-4
MBC
X1-5
RST
X1-6
RX
X1-7
P1.0
X1-8
ADJ_1
ADJ_2
ADJ_3
Remote Control
Wheel + 2 Buttons
2010/52
Wheel
22.02.2011 20:15:37
Sheet: 1/1

Figure 36 IR Transmitter - Schematics

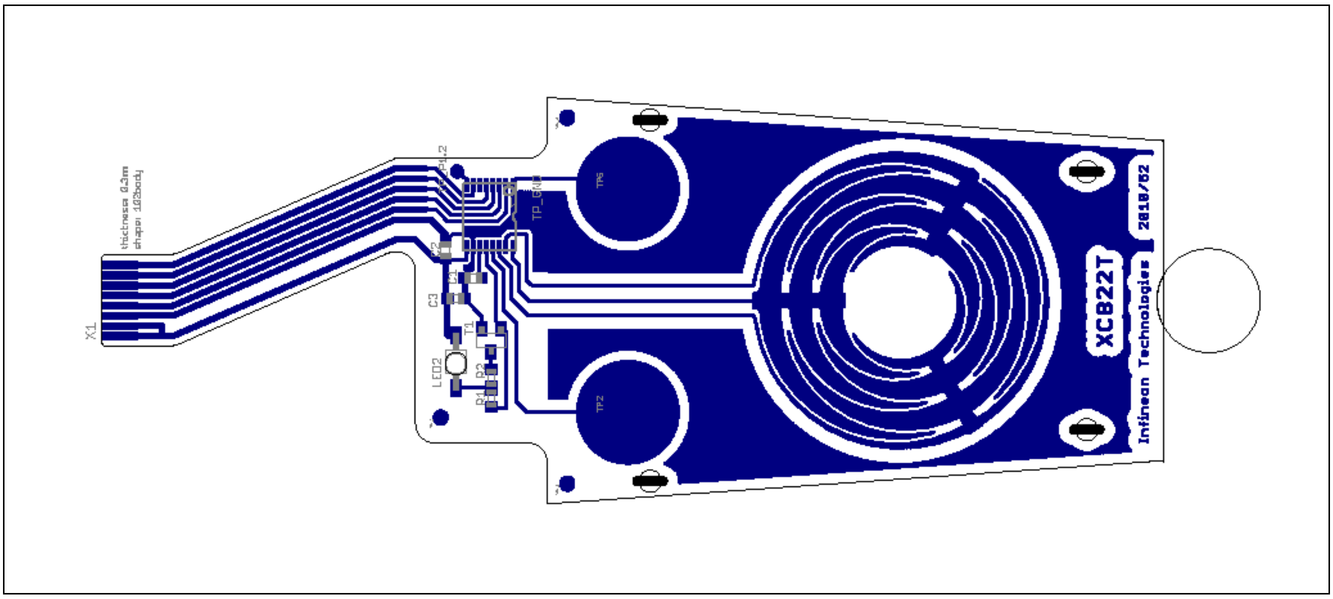


Figure 37 IR Transmitter – Layout (single layer, flexible PCB) and Components

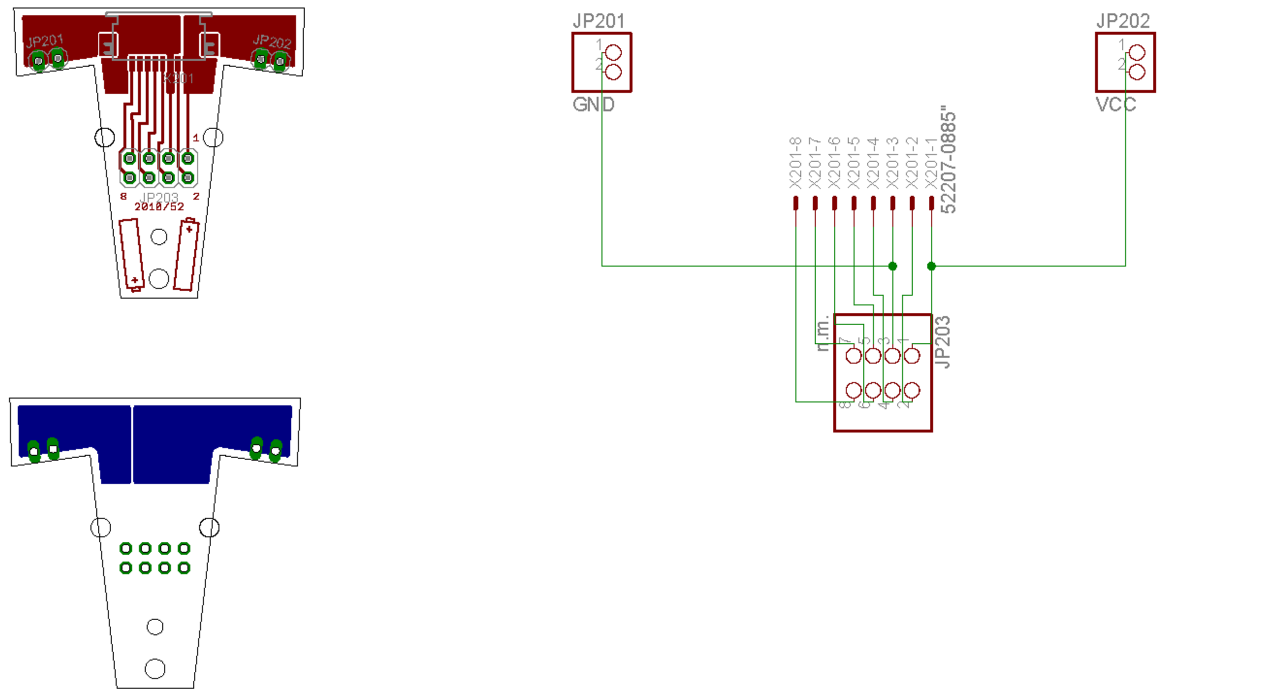


Figure 38 Battery Connector for IR Transmitter

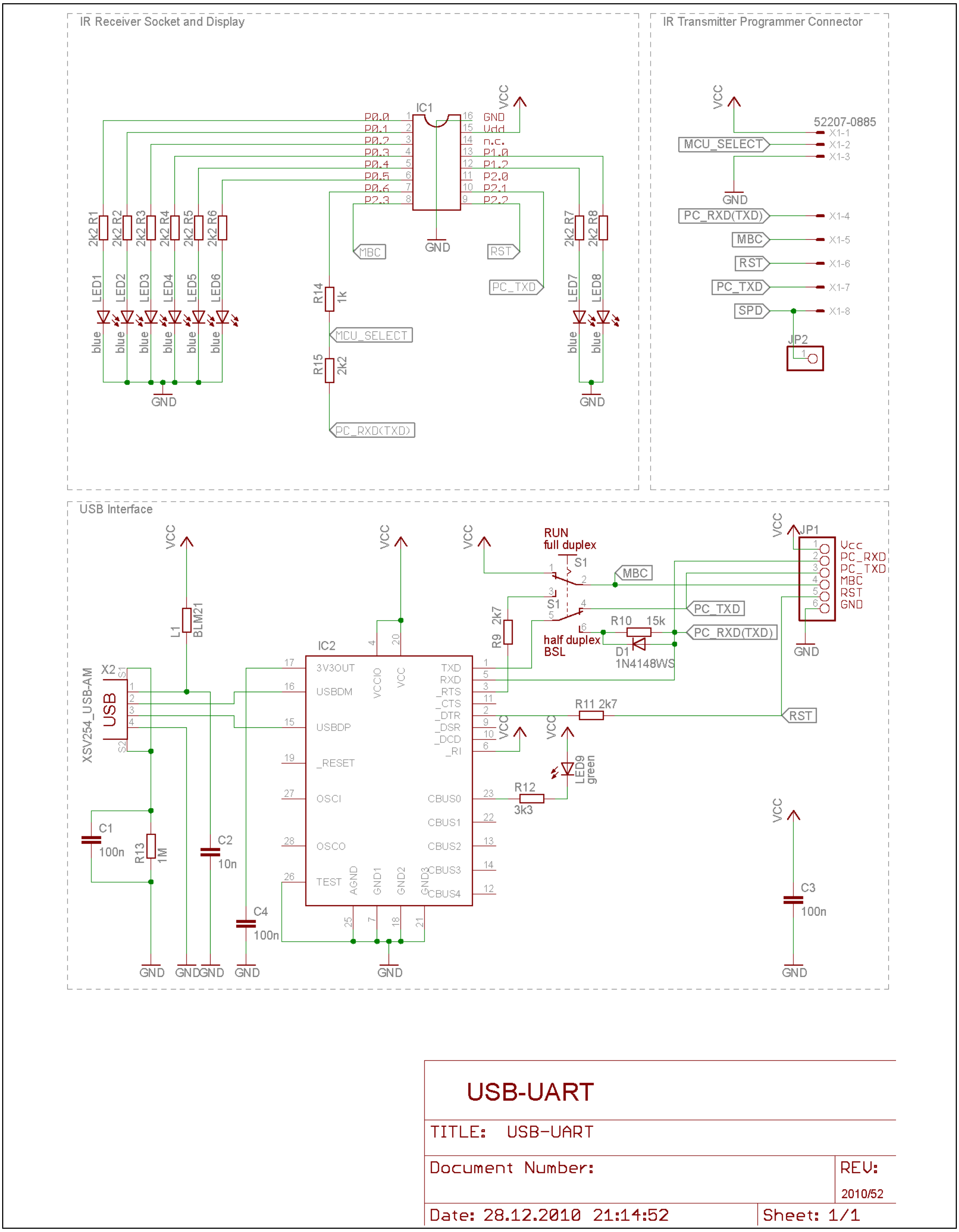


Figure 39 USB Docking Station - Schematics

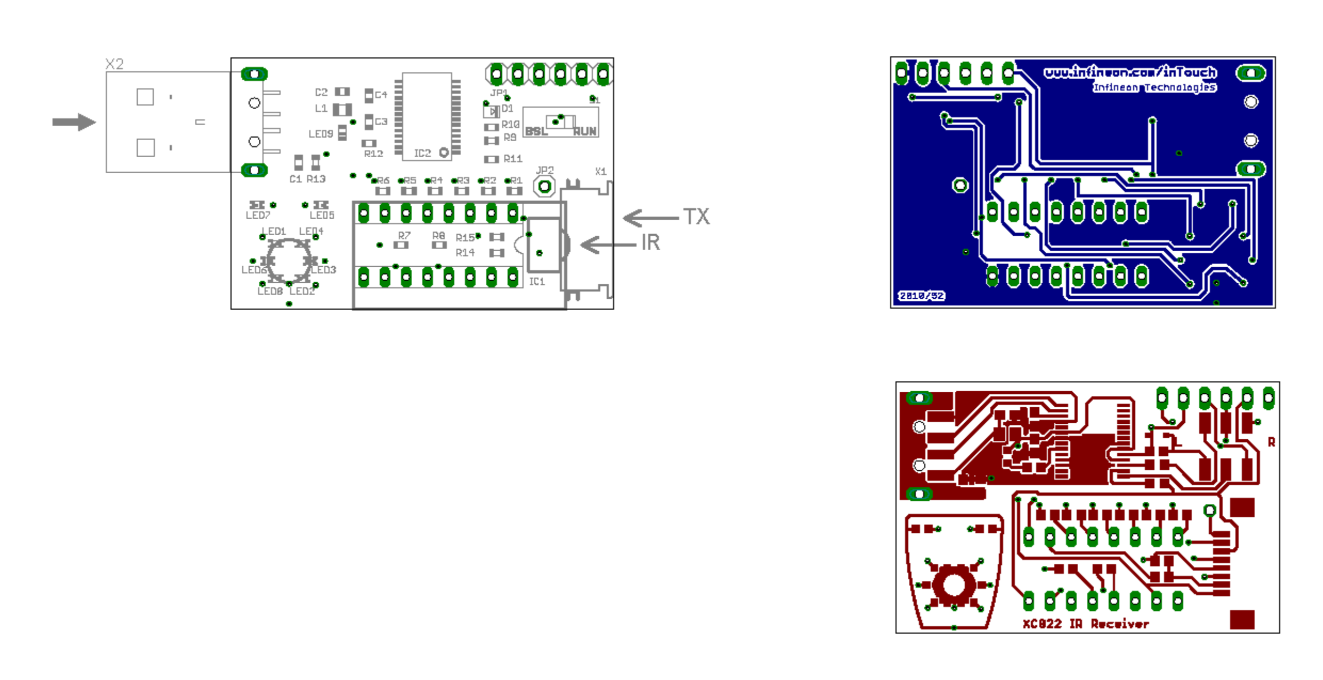


Figure 40 USB Docking Station – Layouts and Components

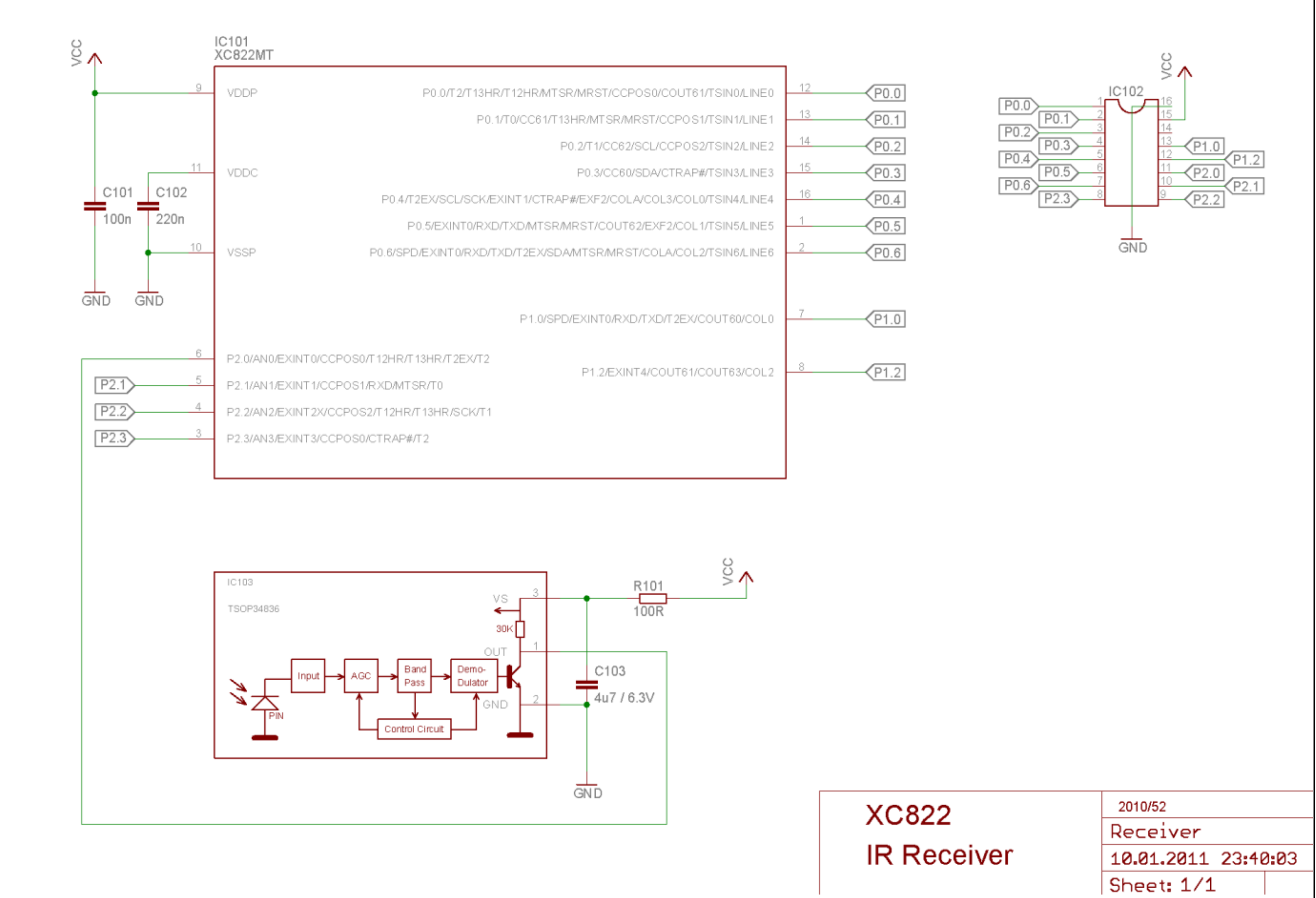

VCC
IC101
XC822MT
VDDP
VDDC
VSSP
C101
100n
C102
220n
GND
GND
P0.0/T2/T13HR/T12HR/MTSR/MRST/CCPOS0/COUT61/TSIN0/LINE0
P0.1/T0/CC61/T13HR/MTSR/MRST/CCPOS1/TSIN1/LINE1
P0.2/T1/CC62/SCL/CCPOS2/TSIN2/LINE2
P0.3/CC60/SDA/CTRAP#/TSIN3/LINE3
P0.4/T2EX/SCL/SCK/EXINT1/CTRAP#/EXF2/COLA/COL3/COL0/TSIN4/LINE4
P0.5/EXINT0/RXD/TXD/MTSR/MRST/COUT62/EXF2/COL1/TSIN5/LINE5
P0.6/SPD/EXINT0/RXD/TXD/T2EX/SDA/MTSR/MRST/COLA/COL2/TSIN6/LINE6
P1.0/SPD/EXINT0/RXD/TXD/T2EX/COUT60/COL0
P1.2/EXINT4/COUT61/COUT63/COL2
P2.0/AN0/EXINT0/CCPOS0/T12HR/T13HR/T2EX/T2
P2.1/AN1/EXINT1/CCPOS1/RXD/MTSR/T0
P2.2/AN2/EXINT2/CCPOS2/T12HR/T13HR/SCK/T1
P2.3/AN3/EXINT3/CCPOS0/CTRAP#/T2
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P1.0
P1.2
P2.1
P2.2
P2.3
IC102
P2.0
GND
IC103
TSOP34836
VS
30K
OUT
GND
Input
AGC
Band Pass
Demo-Dulator
Control Circuit
PIN
R101
100R
C103
4u7 / 6.3V
XC822
IR Receiver
2010/52
Receiver
10.01.2011 23:40:03
Sheet: 1/1

Figure 41 IR Receiver - Schematics

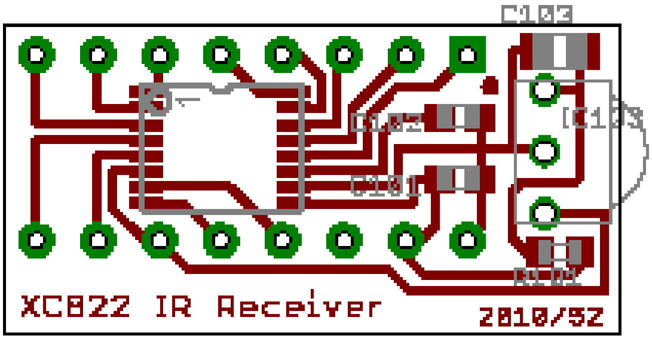


Figure 42 IR Receiver – Layout (Single Layer) and Components

9 References

User's Manual – XC82x; 8-Bit Single-Chip Microcontroller

Data Sheet – XC822/824; 8-Bit Single-Chip Microcontroller

Application Note – *AP08100* – Configuration for Capacitive Touch-Sense Application

Application Note – *AP08098* – Low Power Modes with Periodic/Real-time Clock Wake-up in XC82x/XC83x

Application Note – *AP08101* – Current Consumption in Power saving Modes for Low Power Applications

Application Note – *AP08108* – Programming the BMI value in the XC82x and XC83x products

Link to *XC82x-Series* – www.infineon.com/xc82x

Link to *Solutions for advanced touch control* – www.infineon.com/intouch

www.infineon.com

Published by Infineon Technologies AG