Harvest Electronics

10 PRAGNELL ST, MASTERTON, NEW ZEALAND TEL (+64) 6 370 1991-FAX (+64) 6 370 1993 – www.harvest.com

Radio and GSM DTMF Remote Controller

Connects to an external radio or uses optional plug in GSM module to read eight inputs and control eight outputs at a remote site with full voice response.

The Harvest DTMF controller is available in three versions

1. **Professional GSM version** with both radio and GSM options fitted (includes mains power supply/charger and 1GB SD card)

2. **Professional radio version** with radio control only (includes mains power supply and 1GB SD card)

3. **Amateur radio kitset version** with radio control only. The specially discounted kitset version is only available to licensed ham radio operators (callsign required and shipping must be to call book or QRZ.com address) with a maximum of two per person. Does not include mains power supply or SD card.

Specifications

Voice status of inputs and outputs
Voice dictionary stored on standard SD card
Simple process to add or replace words in any language
Password protected operation – two levels – read status only and read/change status
Four Analog inputs – 2.5V and 25V input range jumper selectable
Four Digital inputs – Internal pull-up resistor – Ground to activate
Dallas One-Wire ® interface for temperature measurement
Four normally open relay outputs – can switch up to 3A loads
Four FET outputs – to switch up to 500mA at 12V
Variable voltage output (DAC) – can supply up to 1A depending on selected voltage
Holtek HT9170 DTMF decoder
Texas Instruments MSP430 microprocessor
Audio and PTT interface for radio connection - isolated
Optional GSM cellphone module
6-18V DC supply
RS232 serial interface for reprogramming the micro

The configuration of the controller can be done relatively easily by the end user with a PC. If you prefer to have Harvest Electronics program the unit for your application we will do that for US$50 and email you the finished file usually within 24 hours. Email info2@harvest.com for this service.

Support

The professional versions come with full telephone help desk and email support and 12 month warranty. Email info2@harvest.com for support or call the (+64) 6 370 1991 during New Zealand business hours. The Ham kitset version has no warranty and is supported only by email from one of the hams at Harvest Electronics. Email info2@harvest.com – no telephone support available for the kitset version. The C source code for the GNU MSP430 compiler is freely available from our web site. Any additional features added by users will be gratefully received and will be included in future releases and software updates. All users (professional and ham radio) agree that the intellectual property in this product belongs to Harvest Electronics and you also agree to have lightning strike you if you make copies of this product for commercial use!

History

The Harvest DTMF controller was originally conceived as a fun Ham radio project but the huge effort involved meant that we would probably never have had the time to complete it after hours. So we have combined the radio and GSM cell phone features into a product which we have developed during work time. We are probably kidding ourselves but we like to think it might have some commercial applications to justify the time and money spent developing it. Peter Munn (ZL2LD) the founder of Harvest Electronics wanted to give something back to the amateur radio hobby that launched his career in electronics so we have made the kitset version available at close to the cost of the components. The PCB was laid out by Gary Anderson ZL2VAR who is a design engineer at Harvest. Support is generally by Glenn ZL2TLD another one of the hams here at Harvest.

Voice Dictionary

The voice response dictionary is stored on an SD card as wave files. Users can add words using any wave recorder using 8 or 16 bit sound files. The files should only have very short silent periods at the beginning and end and should ideally be recorded at consistently high levels. The wave files are stored in a directory called WAVE to overcome the restriction on the number of files which can be placed in the root directory.

We used the Anysound Recorder http://www.any-sound-recorder.com/ for recording the wave file as it has a nice voice operated recording feature to reduce the silence from the beginning and end for files. The free trial version can record files up to 30 seconds long which should work for any words or phrases in this application. The files can be recorded as 8 or 16 bit but must be **MONO.** The sample files in the dictionary are recorded in 16 bit format which sound better than 8 bit. If you record in stereo the microprocessor won't be able to process the data fast enough and the words will come out very slow.

We also used the excellent freeware Wavnormalizer program http://www.linearteam.dk/default.aspx?pageid=wavnormalizer to normalise the wave files and get all the levels consistent.

Words added to the library would normally have a name corresponding to the word(s) in the file. Eg if you added “California” as a word it's a good idea to call it california.wav rather than ca.wav or you will end up confusing yourself when it comes to editing the config file. If you add a file with more than one word like “San Diego” you need to call the file “san diego.wav” and make sure you put the space between “san” and “diego” in the config file.

If you have a word in the config file that the controller can't find on the SD card it will spell out the word. That might be okay for “bilge” but could be a bit painful for “Primary bilge pump”. If you don't want to add words to the library it would be better to use “pump one” in the config file rather than “primary bilge pump” as “pump” and “one” are standard words in the dictionary

The decimal point must be called point.wave because you can't make a file called “..wav” So if an analog input was connected to 12.8 volts and config was set to report volts the software will find and read out the file 12.wav then the file called point.wav then the file 8.wav. If you are doing another language version you will need to redo the voice files in that language but use point.wav for the name of the file containing the decimal point in your language.

The sample files were recorded by Peter (ZL2LD) in his office. His voice is definitely not radio announcer quality and it has a distinctive Kiwi accent which may not suit your application. Feel free to redo the library with a better voice and send us a copy to include with our web site files. You may have suggestions of words which you think would be useful to be in the voice library. If you would like the words to be consistent with the other files and don't want to redo the whole library yourself let us know and we will probably add them for you and email you the new wave files.

Configuration File

The SD card also holds a file called config.txt in the WAVE directory with the voice files. It defines how the controller operates. The config.txt file is a simple text file which can be edited with notepad or any text editor or even Microsoft Word as long as you are careful to save the file as config.txt NOT config.doc. The simplest way to set up a config file is to edit an existing file. There are sample files on the supplied SD card and also on our web site www.harvest.com under the section containing product manuals. Config files must be plain text. Lines can end with a carriage return (CR) or Carriage Return and line feed (CR/LF). If you donlt understand this just use a text editor like Notepad that comes with Windows (find it under programs/accessories) and it CR/LF at the end of each line.

If no config.txt is found the following defaults apply

Status Password   =   ;ie none - if defined this password allows status of inputs and outputs to be read but not changed

Output Password   = 1234 ; this password can read or change all inputs and outputs

Analog inputs all enabled to read volts – eg "Input one 12.8 volts"

Digital inputs read on or off eg "Input 4 off"

Outputs are switched on or off with 1 or 0 eg "Output 1 ON"

DAC is disabled

Temp input disabled

Login Timeout = 120   ; defines how long in seconds before the password expires. This resets every time a valid tone is received.

Key Timeout   =   10   ; After a valid DTMF tone is received if no # or * key is received in this period in seconds the controller clears the command buffer

Push To Talk Timeout   = 500   ; Delay in milliseconds between PTT relay operating and audio being sent. Most radio transmitters take a few hundred milliseconds before they are ready to send audio. For cell phone only operation you can still leave this default or set it to zero for quicker response. For dual radio and cell phone operation you need to leave this at around 500ms.

Pulse Timeout = 500   ; Time of a relay output pulse in milliseconds when using the pulse option

reply delay   =   1   ; Time in seconds after a valid command is received before a reply is sent. This can be useful if you wish to send multiple commands before a response is received. If multiple commands in quick succession are not required you can set this to zero.

Anything after a semicolon (;) is a comment and has no effect on the operation of the controller. Spaces are ignored in commands so you can make the config easier to read

Harvest DTMF decoder default dictionary –Contains about two hundred common words and is available for free download from the www.harvest.com web site under product manuals.

Setting up

Jumpers

The *Option* jumper is presently unused – available for future software improvements.

The *Program* jumper, when installed, allows the Strap.exe program to send new software to the microcontroller.

The *LED Enable* jumper enables status LEDs on the relay outputs. It can be removed to reduce power consumption if required.

The *Analog Input Range* jumpers, when installed, configure each analogue input for 25V full scale. If you require the 2.5V input range, remove the jumper on the appropriate port and alter the configuration file.

Power

The unit requires a DC power supply of between 6V and 18V. The maximum current required depends on third party equipment attached to the decoder board. The board itself typically uses 1mA but needs a power supply that can deliver at least 200mA when the cellphone is transmitting.

Serial port

The serial port is used to reprogram the controller with newer versions of firmware which can be downloaded from the harvest web site www.harvest.com under support. You will also need to download the program called strap.exe to program the units. The serial port can also be used to debug what is happening under normal operation with a PC and terminal software running at 9600, N, 8,1 . The micro sends debug information while the unit is operating.

SD card

The SD card is supplied pre-programmed with common words and phrases and a sample configuration file. The user may replace any of the recordings and can add additional ones. The file name of the recording is referenced in the configuration file.

The file name of each voice file on the SD card is normally the word or phrase that is to be spoken. Spaces are permitted in filenames. The files are all in one directory (WAVE) on the SD card. This is because the root directory is limited to 512 entries, and having the files in a sub directory circumvents this limit.

SIM card

If the GSM cellphone option has been purchased, the user will need to install a SIM card supplied by a local GSM network operator. We generally use a pre-paid SIM as the DTMF decoder phone typically only receives calls. In the future there will be an option to send text (SMS) messages and voice messages for alarm outputs.

Analog inputs

The four analog inputs can be jumper selected for 0-2.5 volts or 0-25 volts. Sensed voltages can be spoken directly or they can be scaled, offset and given a unit name. E.g., a wind speed sensor that produces a voltage proportional to speed could provide a reading in knots or feet per second or kilometres per hour, depending on the scaling factor applied and the 'units' voice message attached to the input. An ultrasonic sensor could have a scaled output as well as an offset for distance. These four inputs are addressed as 41-44

Digital inputs

The four digital inputs are intended for door switches, float switches or relay contacts. They have an internal pull-up resistor and are considered to be 'active' when grounded. These four inputs are addressed as 45-48 and will accept an input up to the voltage of the supply

One-Wire ® interface

The Dallas One-Wire ® interface is intended for the Dallas 18B20 temperature sensor. With appropriate software support, this interface could be used for any number of one-wire ® devices. These are available from Harvest – including a version in an outdoor radiation shield that we use for our weather stations. This input is addressed as 81

This can be remote from the unit with a length of any two-conductor cable, with attention paid to polarity. The voice output can be set to report the temperature in Celsius or Fahrenheit .

Relay outputs

These relays have normally open contacts and can switch up to 30 volts at 2 amps. The contacts can be closed, opened, toggled or pulsed by commands as defined in the configuration file. These relays outputs are addressed as 61-64

FET outputs

These four FDV303N N-channel mosfets are intended to switch up to 12V at 500mA. Their absolute maximum rating is 25V at 680mA. These outputs are addressed as 65-68

DAC output

This DAC (digital to analogue converter) can be used to generate a voltage to operate remote equipment at 1-12 volts. At 12 volts the DAC can output up to 1 Amp. The DAC output circuitry can only dissipate 2 watts so the maximum current must be derated below 12 volts roughly by the formula I (mA) = 2000/(Vsupply – Vdac) with a maximum of 1000mA. As an example with a supply of 13 volts and a DAC output of 3 volts it can output 200mA. The DAC output voltage is set using the address 31, followed by the required voltage in hundredths of a volt. **31530#** will set the output to 5.3V. **3162#** would give 0.62 volts. If a voltage greater than the supply voltage is selected, the unit will provide the supply voltage. **310#** will turn the DAC output off. **31#** will read the DAC output without changing it. The DAC default value can also be set in the config file but any changes made by **31** commands will override the config settings until the next power reset.

Uses for this might include switching on a soil moisture probe at 3 volts then reading the level in one of the analog inputs. The DAC could be switched off after the reading is taken to save power.

Radio interface

There are separate connections for audio from the radio to the decoder (Audio In), audio from the decoder to the radio (Audio Out) and a relay contact for transmitter activation (PTT). The audio in and out lines are isolated from each other so you can even use the PTT line to operate radios that require the grounds of the MIC and EAR to be joined to operate PTT.

There is also a speaker output (Spkr). This is intended to be connected to a small speaker for local monitoring and testing. There are two adjustments labeled ‘audio out’ and ‘spkr’ on the board for setting the audio levels.

We expect these units to be interfaced to Ham, Commercial VHF/UHF or UHF CB (FRS/PRS) radios. There are detailed connection configurations for a variety of radio interfaces avail on our website under support http://harvest.com/support.htm

Cellphone Option

The cell phone used is a Motorola g24 quad band GSM engine which will operate anywhere in the world that has GSM coverage. You will need to get a SIM card to put in the phone. Most users get a cheap prepay SIM card which will often last for a year or more without adding any funds. The cell phone is programmed to answer on the first ring and disconnect when caller hangs up.

Cellphone antenna

If the cellphone option has been installed, an antenna appropriate for the GSM band used in your location must be attached. The cellphone modules supplied work on 800, 900, 1800 and 1900 MHz GSM bands. The antenna connector on the unit is an SMA. Cellphone equipped units will be supplied with a PCB type antenna which is adequate in primary coverage areas. Fringe areas or other frequencies of operation will require an antenna suitable for the installation. Harvest Electronics can also supply external mag mount antennas.

Configuring the unit – software

Reprogramming micro flash – not normally required unless new features are needed for your application

To load any software updates into the microcontroller you will need the Strap.exe program and the dtmf.hex object code file. These are available from our website under product manuals. http://harvest.com/support.htm Fit a jumper on the *Prgm* pins, attach the serial cable to your computer and issue the command ***strap –c dtmf.hex***

The strap program assumes com1. ***strap -?*** will show the options available to change the com port.

Remove the *prgm* jumper when the code has been loaded to restore normal operation.

Configuration parameters

Login Timeout is the number of seconds the unit remains unlocked when the password is given. This timer is reset when a DTMF digit is received. When the timer expires the password will be required again.

Key Timeout is the number of seconds a partial command is remembered for. When this timer expires the scratchpad is erased. This is to prevent unexpected behaviour.

Push To Talk Timeout is the delay between the PTT relay turning on and the voice response beginning.

Pulse Timeout is the duration of a pulse on an output.

Reply Delay is the pause between receiving a command and transmitting the response.

Status Password is the password for reading inputs and outputs but will not allow changing the state of any outputs. Default = no password required

Output Password is the password that allows full control. Default = 1234

Passwords may be any length up to eight digits long. They must consist solely of digits and leading zeros are not permitted. Ie 0123 would be treated as 123

ADC Range values should match the jumpers on Analog Inputs 1 to 4. These numbers are used to get the right scale when values are reported.

Commands

Each input and output has a line in this section to define the manner in which commands are executed and values reported. Let's examine the sample config.txt file line by line.

Sample Config.txt file

```
Output Password   = 8532

Input  1 = “battery” ; connected to rechargeable battery supply

Input  5 = “Mains” ; connected to 12 VDC plug pack to indicate if the mains power is on or off

Output  1 = “pump” “1”

Output  2 = “pump” “2”
```

This config file will use the defaults for all parameters not listed

Examples of Config file entries

Input: 1 = "battery"

This is a simplest and default type of input which measures a DC voltage and reports in volts. For example the supply voltage can be measured if you attach a wire from the positive battery lead to one of the analogue inputs. In this example we are using input 1. If the battery voltage was 13.2 volts then sending the DTMF command **41#** will result in a voice response of “Battery Thirteen Point Two Volts”.

If no entry was made in the config file for this input it would read “thirteen point two volts” without the word “battery”. If there is no entry for an input or output the status will not be announced with a read all status command **49#** but can still be read individually with the **41#** command.

If you are using the default setting for an input that you will have connected it is a good idea to add an entry in the config file. The line “input 1 =” will suffice. The software ignores all spaces in commands so you can make the config file easier for you to read by putting in some spaces as in the examples. You can add comments to help you remember what things are for by putting a semicolon (;) before the comment. If a comment goes onto the next line you need to put another semicolon on that line as well.

****

Input: 2 = "speed",   Multiply = "0.0345", Offset = "0.5", Unit = "kilometers per hour"

This is an example of a more complex command where the input voltage is scaled and offset to produce a meaningful output.

The value read from input 2 is multiplied by 0.0345 then the offset of 0.5 is added. The voice response will be of the format “Speed Forty Three Kilometres Per Hour”. The word “speed” is probably not required so you could leave that out.

Input: 2 = ,   Multiply = "0.0345", Offset = "0.5", Unit = "kilometers per hour"

Note if no name is to be used for the input you still need a comma before the next item

****

Output: 1 = "motor",   ON   = "on”,   OFF = "off"

When the DTMF command **611#** is sent the output 1 relay will close and the voice response will be “Motor on”

When the DTMF command **610#** is sent the output 1 relay will open and the voice response will be “Motor Off”

‘OFF’ and ‘ON’ do not need to be explicitly stated here. They are the default values.

So the config program could have been

Output: 1 = "motor"

****

Output: 2 = "pump" "1"

More than one voice recording file can be used. When the DTMF command **621#** is sent the Output 2 relay will close and the voice response will be “Pump One On” and when the DTMF command **610#** is sent the output 2 relay will open and the voice response will be “Pump One Off”. The command **61#** will read the current status of output 1 without changing the status.

****

Output: 3 = "door",   PULSE:500 = "unlatched",   OFF = "locked"

When the DTMF command **631#** is sent the relay on output 3 will be close for half a second and the voice response will be “door unlatched”. The **PULSE:500** defines the duration in milliseconds. **PULSE** on its own will use the *Pulse Timeout* default value of 500ms. Sending the command **630#** will have no effect as the output is only a pulse output but it will return the voice response “door locked”

****

Output: 4 = "Siren", ON = “DO NOT SEND”, OFF = "off"

“DO NOT SEND” is a special name which is not a voice file. It tells the controller not to send any voice response for that input. This is useful you want to send another command immediately afterwards or if the voice response is not wanted for some reason.

****

Temperature: , C

The **C** here converts the temperature reading into degrees Celsius. An **F** will convert to Farenheit.

81# reads the temperature of a Dallas 1-Wire temperature sensor connected to the temperature input.

Alternatives

Temperature: “temperature”, C, “degrees ” would announce “temperature 10 degrees”

Temperature: “temperature”, F, “degrees” would announce “temperature 50 degrees”

Temperature: “temperature”, C, “degrees Celcius” would announce “temperature 12 degrees celcius”

****

ALARMS

The ability to send SMS and voice alarms is under test but not available for general release yet

Operation

If a password is required on auto answer, the GSM phone announces “please enter password”

Passwords are variable length but must end with an asterisk (*) eg 1234*

If the password is defined in config.txt as blank then no password is required

It is possible to set a read only password so some users can read the status but not change outputs. Normally the read only (status) password would be left blank as users checking the status cannot do any damage.

After a valid password has been entered the system will accept commands for the password timeout period. Every time a DTMF digit is received the password timeout is restarted.

All other commands must end with a hash (#)

The password and single or multiple commands can be sent as one string eg **1234*611#** will send the password and turn the output 1 relay on. The password can be sent with every command if desired. This is useful for storing commands in a cell phone or radio memory. In one DTMF memory you could store **1234*611#** to turn something on and in another memory you could store **1234*610#** to turn it off even though the password was not really required if the password was still active. When a command is included straight after a password the message “password accepted” is not announced unless the password is not correct or not heard clearly – then the controller will announce “password rejected”.

Most cell phones can store DTMF codes as part of a stored phone number. You may have to eneter some pauses in the stored number (usually entered as a comma for one second pause. You can add more commas until everything works correctly. It is usually easier to put plenty of commas in to start (say 5) then remove them one by one until it stops working reliably then add one back in. When passwords and commands are stored in a DTMF memory it is a good idea to put a # in front of the string eg **#1234*611#** will send a password of 1234 then turn output 1 on. **The extra # at the beginning helps the DTMF decoder to not miss any digits in the high speed string.**

Radio

Start transmitting and enter the password (if configured) followed by an asterisk (*). The controller will remain unlocked until no commands have been received for the timeout period configured in the config.txt file. The default timeout is three minutes.

DTMF Commands

All commands must end with a hash (#).

To read an input, prefix the input number with a Four. To read input 3, key **43#.**

Digital inputs 1 to 4 are input numbers 5 to 8.

The Dallas One-Wire temperature sensor (if installed) can be read using **81#**.

To set an output, prefix the output number with a Six. The FET outputs 1 to 4 are output numbers 5 to 8. Key **620#** and relay output two will be turned Off. **671#** will turn FET output 7 On. Optionally a **one** can toggle or pulse the output. This is set in the **config.txt** file.

The current state of an output can be read. **63#** will report the status of relay output 3.

69# will report the status of all outputs that are defined in the configuration file.

Inputs are prefixed with the number 4 eg inputs 1-8 are addressed as 41 to 48. We chose 4 for inputs because on a telephone keypad the digit 4 also corresponds to the letter “I” for “Input”

Outputs 1-8 are prefixed with the number 6 eg 61 to 68. We chose 6 for outputs because on the telephone keypad the digit 6 also corresponds to “O” for “Output”

INPUTS

Read input 1 by sending 41#, input 2 by sending 42# etc

Read all defined inputs (ie have entry in Config.txt) by sending 49#

Dallas 1-wire temp sensor is read by sending 81#. We chose 8 for temperature because on a telephone keypad the digit 8 also corresponds to “T” for “Temperature”

OUTPUTS

Normally 1 turns an out on, 0 turns it off eg 611# is he command to turn output 1 on

Optionally a 1 can toggle the output , send a pulse (programmable length – default 0.5 second) or operate the output as long as the tone is sent (momentary). These options are set in the Config.txt file on the SD card. If the toggle option is chosen sending a 0 will always turn the output off. When momentary or pulse options are set sending a 0 has no effect.

610# turns output 1 off 61# reads the status of output one but doesn’t change it

69# reads the status of all valid outputs (those that have entries in Config.txt)

DAC (Digital to Analog Converter)

The value of the DAC is set in the config.txt. The DAC can be turned on and off as an output by the address 31. We chose 3 for the DAC because on a telephone keypad the digit 3 also corresponds to “D” for “DAC”

If a wrong key is pressed end that sequence by pressing * which will interpret the entry as an invalid password

To help you remember commands

We chose **3** to prefix **DAC** commands because that corresponds to “D” = DAC on a telephone keypad

We chose **4** to prefix **Input** commands because that corresponds to “I”=Input on a telephone keypad

So **41# reads** input 1, **42#** reads input 2 etc Special command **49#** reads all inputs specified in the config file and announces the status of all outputs specified in the config file. This command is useful for quickly checking the status of a remote site

We chose **6** to prefix **Output** commands because that corresponds to “O”=Output on a telephone keypad

A “1” turns an output on and a “0” turns it off

So **61#** reads the status of output 1, **62#** reads the status of output 2 etc

And **611#** closes output relay 1 and **610#** opens relay 1

69# will report the status of all outputs defined in the Config.txt file

We chose **8** to prefix **Temperature** commands because that corresponds to “T” on a telephone keypad

So **81#** reads temperature 1

PASSWORD

The password can be sent in the same string as control commands eg for a password of 1234 the following sequence 1234*611# will enter the password and operate output 1

The password can be sent with every command which is not necessary but is useful when the command is programmed into the phone or radio.

DTMF sending speed

The DTMF controller will accept digits from 50ms to 500ms which roughly corresponds to the speed of an autodial string and manual sending respectively. For numbers stored for high speed sending prefix the commands with a # so the decoder doesn’t miss any digits. For manual sending from a keypad there is no need to send a # before a password or command.

VOICE RESPONSE

Normally all output control operations are confirmed by voice but there is an option to not send a voice response for either the 1 or 0 state when you may not want to the delay of a voice response before sending another command. See the Config.txt file format examples in this document. You can still request the status by voice response.

The name of each input and output is defined in the Config.txt file. If a name is chosen for which there is no wave file stored on the SD card in the WAVE directory the controller will spell out the word. This can be pretty painful for long names so you need to either add your own wave file or abbreviate the name.

HAM KITSET DOCUMENTATION ADDENDUM

Support

For the Ham version kit, email info2@harvest.com . Your requests will be responded to on a best-efforts basis by one of the ham staff at Harvest.

Kitset documentation

The ham version kitset will be supplied with all of the surface-mount components professionally installed, and a detailed visual inspection will be undertaken before the board is shipped. This is because placing and reliably soldering these super-tiny components is a process fraught with peril. We would much rather do the tricky stuff for you and leave you with the easy stuff to do. The robots do a fine job. You can do the thru-hole stuff. We know you can. Hot irons, clean tips and leaded solder.

There will be a plastic bag included which will contain the following parts.

- DC Power plug and socket
- A strip of pin headers and seven jumpers
- Screw terminals
- 600 ohm audio isolation transformer
- five relays
- DB9 socket
- 220uF 50V electrolytic capacitor
- Ribbon serial cable·

The recommended assembly sequence is as follows

1. Pin headers
2. Power socket
3. DB9
4. Transformer
5. Screw Terminals
6. Relays

7. Capacitor

Any issues with the assembly process are best addressed via an email to info2@harvest.com.

A detailed assembly procedure is available on our website with the DTMF decoder manuals.
http://harvest.com/support.htm

In the event of a failure to get the kit operating, we can offer a technical resolution service. You will need to document the exact issue and return the defective board to us. If we determine the issue to be a manufacturing defect, there will be no charge and we will return the product to you freight paid. If, however, we determine the problem to be your ham-fisted assembly technique, we will, regrettably, have to charge you a nominal fee of US$40 to repair the board and return to you a working board. The assembly you are expected to undertake is pretty easy – you should have no problems.

All of the schematics, board layouts are available under “support” at www.harvest.com so we hope that you will be able to resolve any issues for yourself. The source code is available on request for those who would like to add extra features. Replacement discrete components, should you need them, will generally be available for purely the cost of postage. We can assist you (via email) in determining the likely failure cause.

You will need to supply your own SD card and download the contents from the support page at www.harvest.com

FCC ID

If you require an FCC ID you could change the “password accepted” message with a file that included a callsign or alternatively one or more inputs or outputs could have a callsign as part of the name

Eg ***Input: 1 = “ZL2LD" "battery"*** When input 1 was read it would announce “ Z L 2 L D battery 12.8 volts”