Gippstech is a technical symposium held each year at the Churchill campus of Federation University during the month of July. Whilst focusing mainly on VHF weak signal and microwaves there is much in the program for the ardent home constructor.

Of equal value to the structured content is the ability to converse with the like-minded over lunch or coffee. The following project sprang from one such encounter.

A Simple CAT Controller for the FT817

The FT817 is a great rig for VHF contesting either as a stand alone transceiver or the IF for a microwave transverter. With so much frequency hopping the small buttons on this rig and deteriating eyesight can lead to some frustration. In the case of VK3MQ there are six possible IF combinations and a change from USB to LSB for the 2.4GHz transverter. This situation has arisen partly by available hardware but also by the shift of an allocation at 3.4GHz.

The Gippstech coversation suggested an Arduino Nano could be used to control the FT817 via the rear panel ACC port. On returning home a search of the internet for a possible "sketch" found a pic based controller and this was explored for a time.

The original article can be found at:

http://www.creative-science.org.uk/articles/JPH%20RadCom%20Sept%202015%20FT-817%20controller.pdf

The source code for this project was made to work with the available assembler but before placing an order for the pic from an interstate supplier the idea of an Arduino version was reconsidered.

A further search of the internet revealed an interesting library written for the FT857. Whilst not completely compatible with the FT817 the differences proved to be minor and made the Arduino option an easy exercise.

The work by VE3BUX:

http://ve3bux.com/?page_id=501#requirements

Armed with the library and a book on Arduino basics the following test sketch was produced.

```
#include <SoftwareSerial.h>   // to impliment the software serial port

#include <FT857D.h>  //VE3BUX library

FT857D radio;          // define "radio" so that we may pass CAT commands

 const int SIX = 6;          // 6m button on pin D6
 const int TWO = 7;          // 2m button on pin D7
 const int SEVENTY = 8;       // 70cm button on pin D8
 const int TWENTYTHREE = 9;      // 23cm with 145MHZ IF on pin D9
 const int THIRTEEN = 10;       // 13cm with 145MHZ IF & LSB on pin D10


void setup() {

 pinMode(SIX, INPUT); // define pins connected to the buttons as inputs
 pinMode(TWO, INPUT);
 pinMode(SEVENTY, INPUT);
 pinMode(TWENTYTHREE, INPUT);
 pinMode(THIRTEEN, INPUT);
 radio.begin(9600);   // start the software serial port at 9600 baud see NOTE

}

void loop() {
 if (digitalRead(SIX) == LOW) {
   delay(25);
   radio.setMode("usb");    // set mode to USB (note these mode names are NOT case
sensitive!)
   radio.setFreq(5015000);
 }

 if (digitalRead(TWO) == LOW) {
 delay(25);
 radio.setMode("usb");
 radio.setFreq(14415000);
 }
```

```
 if (digitalRead(SEVENTY) == LOW) {
 delay(25);
 radio.setMode("usb");
 radio.setFreq(43215000);
 }

 if (digitalRead(TWENTYTHREE) == LOW) {
 delay(25);
 radio.setMode("usb");
 radio.setFreq(14515000);
}

 if (digitalRead(THIRTEEN) == LOW) {
 delay(25);
 radio.setMode("lsb");
 radio.setFreq(14485000);
 }

}
```

For the benefit of those not used to reading sketches the setup involves five momentary switches connected to pins d6, d7, d8, d9 and d10 in this case with external pull up resistors. When a button is pushed the pin goes low and two instructions (mode and frequency) are sent to the radio via the software serial port (pin d3=rx pin d2=tx).

Pins d3 and d2 are defined as the RX and TX pins by the file FT857D.cpp

Depending on the button pressed the radio will switch to:

50.150MHz USB, 144.150MHz USB, 432.150MHz USB, 145.150MHz USB or 144.850MHz LSB.

As with anything involving software many adaptations are possible. For instance an FM channel or HF frequency could be added. The remaing data pins could be made outputs and used to switch the FT817 between antennas or transverters it is really up to the builder's imagination.

NOTE: The Arduino Nano cannot produce a baud rate of 4800 which is the default for the FT817. Menu 14 will need to be set to 9600.

The prototype was built on vero board using an Arduino Nano bought locally at an electronic parts store.
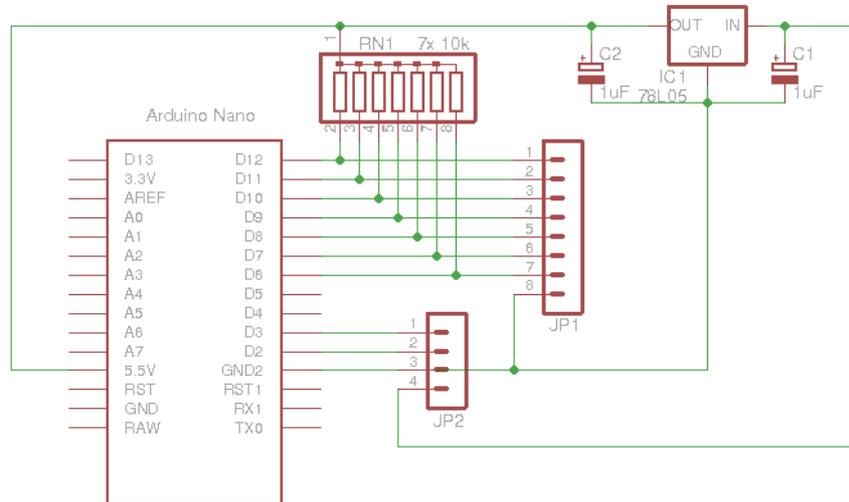

Figure 1 The circuit diagram with expanded switching.
Note : JP1 connects to the switch board and JP2 connects to the FT817
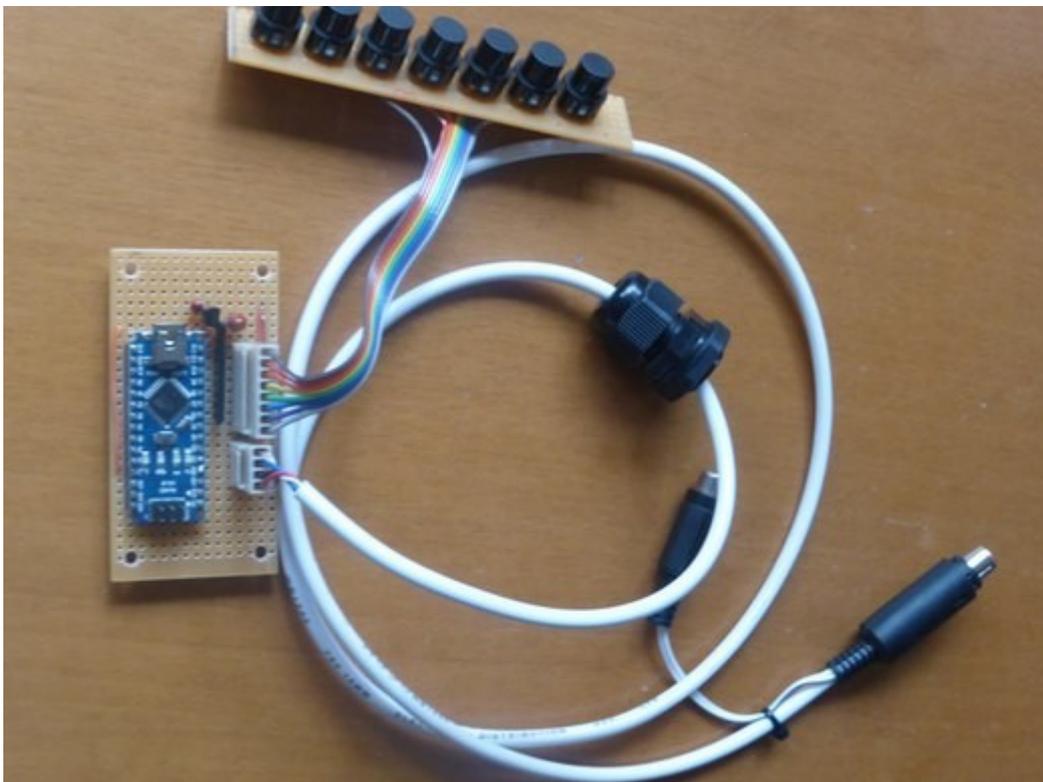ACC socket

Figure 2 The final product.

Acknowledgements

Thanks go to Lou VK3ALB for the original idea and Peter VK3APW for the coversation over a pizza at the Gippstech Sunday lunch.

A Bluetooth interface for the FT8*7 range of transceivers.
http://k6vhf.com/simple-bluetooth-cat-interface/
https://www.yo3ggx.ro/btcat/FT8x7_DIY_Bluetootth_CAT_interface_v1.pdf

A GPS locked 10MHz standard.
http://www.ik0otg.net/index.php?option=com_content&view=article&id=53&Itemid=59&lang=en

Playing with the geometry of a dipole.
http://www.qsl.net/kk4obi/Center-fed%20Bent-end%20Horizontal.html

All the efforts of one ham
http://lea.hamradio.si/~s57nan/ham_radio/

A CAT interface for Icom transceivers
http://www.ka1mda.org/ham/homebrew/civ/index.htm

How does an SWR meter work?
https://www.google.com.au/url?sa=t&rct=j&q=&esrc=s&source=web&cd=14&cad=rja&uact=8&ved=2ahUKEwjn_ZqAj_vcAhWGu7wKHXTIDSoQwqsBMA16BAgEEAQ&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DbyF1FLdbUiA&usg=AOvVaw06QTZKBbcEpNh_sYayr0qW

http://kambing.ui.ac.id/onnopurbo/orari-diklat/teknik/arrl/using-equipment/5904024.pdf

http://www.parc.org.za/attachments/projects/Swrmeter%20Project.pdf

Finally something to think about:

"I am thankful to all of those who said NO. Because of them I did it myself."

Albert Einstein

73's
Rob
VK3MQ