Functional Description of PIC16F877A Functions and Interfaces to GBT RFI Monitor Station Electronics Division Technical Note No. 208

J. R. Fisher & Carla Beaudet

April 26, 2005

1 Introduction

Figure 1 shows a block diagram of the RF components in the GBT RFI monitor station as planned when this document was written. The station includes up to four, two-axis Yaesu antenna positioners, five RF transfer switches for filter in/out selections, and two SP5T RF switches for antenna selection in each of two RF signal paths back to the lab via optical fiber. The positioners and switches are controlled remotely using a 40-pin MicroChip 16F877A microcontroller at the monitor station that receives commands via RS232 and translates them into hardware control logic. The microcontroller connections to the modules that it controls in the monitor station are shown in Figure 2. This note describes the conventions adopted for the microcontroller and its connections to the hardware to be used in writing the microcontroller assembly language program. Detailed descriptions of the PIC16C877A can be found in two product references: “PIC16H878XA Data Sheet” and “PICmicro Mid-Range MCU Family Reference Manual.” These are available under ’Datasheets’ for the 16F877A device on the www.microchip.com web site or as pic16F877.pdf and midrange_mcu_ref.pdf files in my project directory.

The microcontroller clock is generated by an external 10-MHz crystal. This produces a single instruction cycle time of 0.4 microseconds. The program flash memory can be programmed in the circuit through pins 36, 39, and 40 using the CCS ICD-S40 in-circuit programmer.

Interface to the RS232 serial line is via a DS14C232 IC, shown in Figure 3, which converts RS232 logic voltage levels to TTL levels accepted by the 877A. In this application the RS232 signal will be carried on a multi-mode optical fiber by way of an RS232 optical modem. At the lab end of this fiber will be a PC that sends commands to and receives monitor information from the 877A.

The four antenna positioners have a common interface to the 877A as shown in Figure 4. This module is connected to the six most significant bits of register port RD. One of the four positioners is controlled by putting its address on lines A0 and A1, and up/down or CW/CCW motion is commanded by pulling one of the U, D, C, or CC lines low. The azimuth and elevation positions are read by 877A A/D converters AN0 and AN1, respectively.

Each of the five filter selection transfer switches is controlled with a single bit in port resister RB. Each of the two SP5T RF switches is controlled with three bits in port register RC, where the three bits form an address to one of the five switch positions.

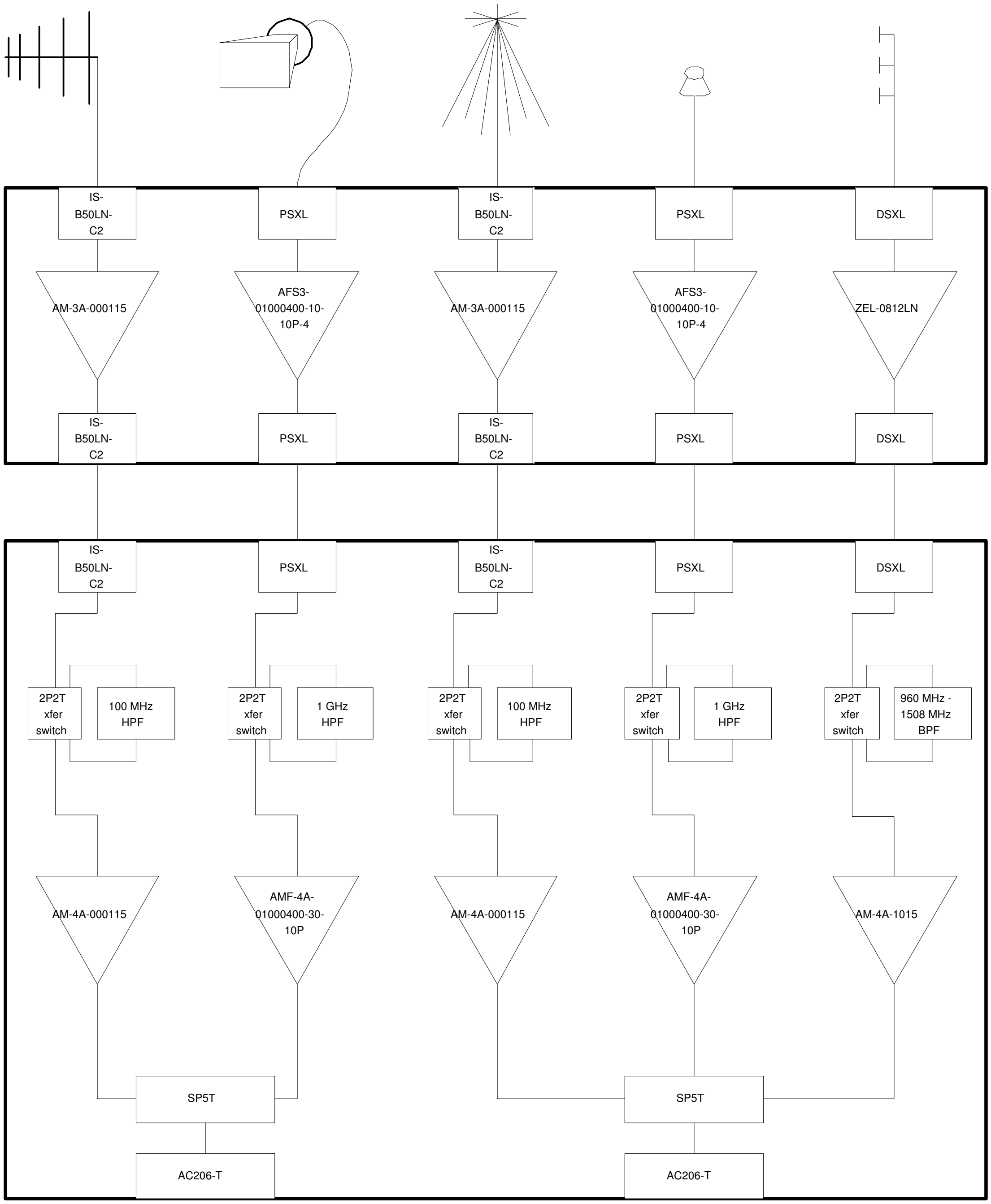

Figure 1: RF components block diagram of the full GBT RFI Monitors station as planned.

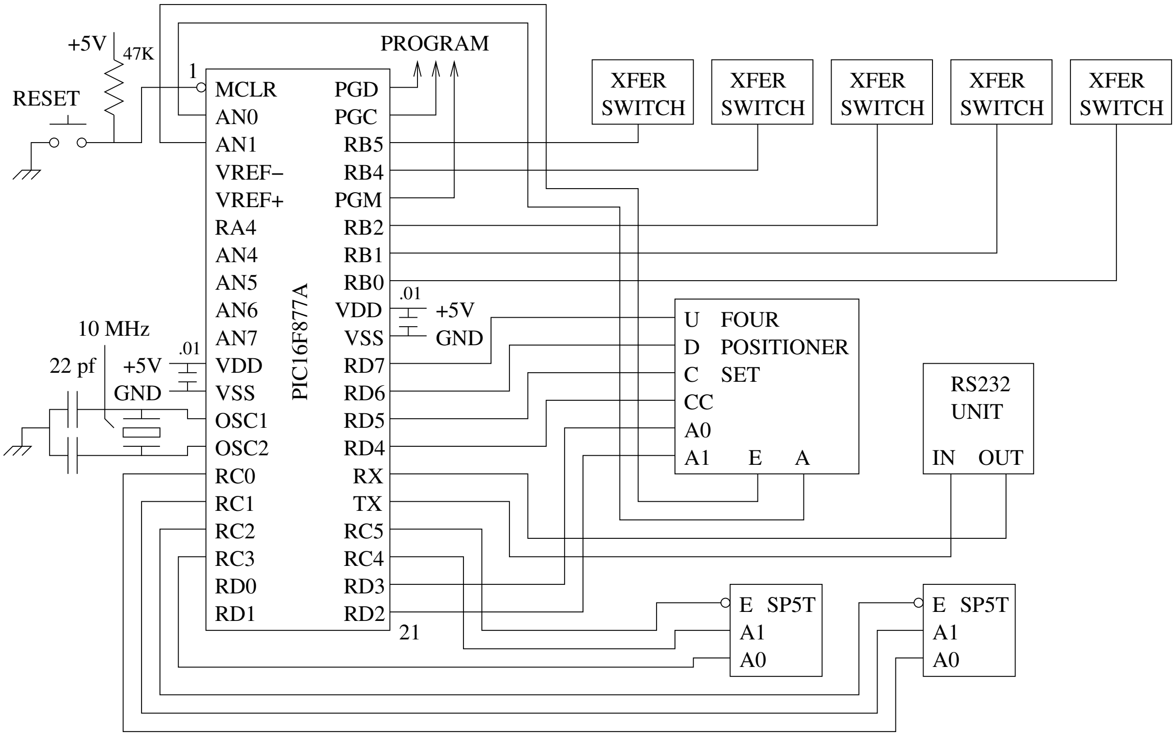

Figure 2: Schematic diagram of microcontroller connections to the antenna positioners and RF switches that it controls. More detail on the positioner and switch controllers are shown in Figures 3 through 5.

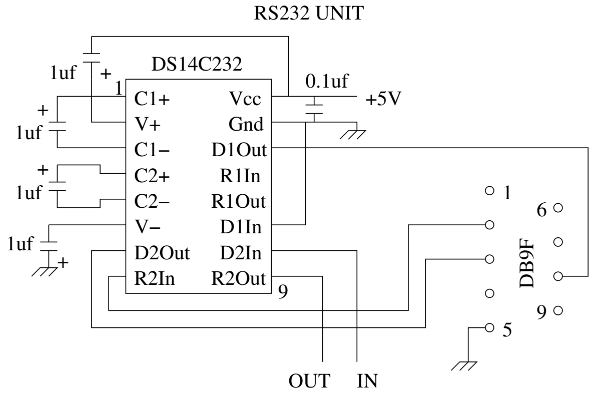

Figure 3: Schematic diagram of RS232 interface circuit.

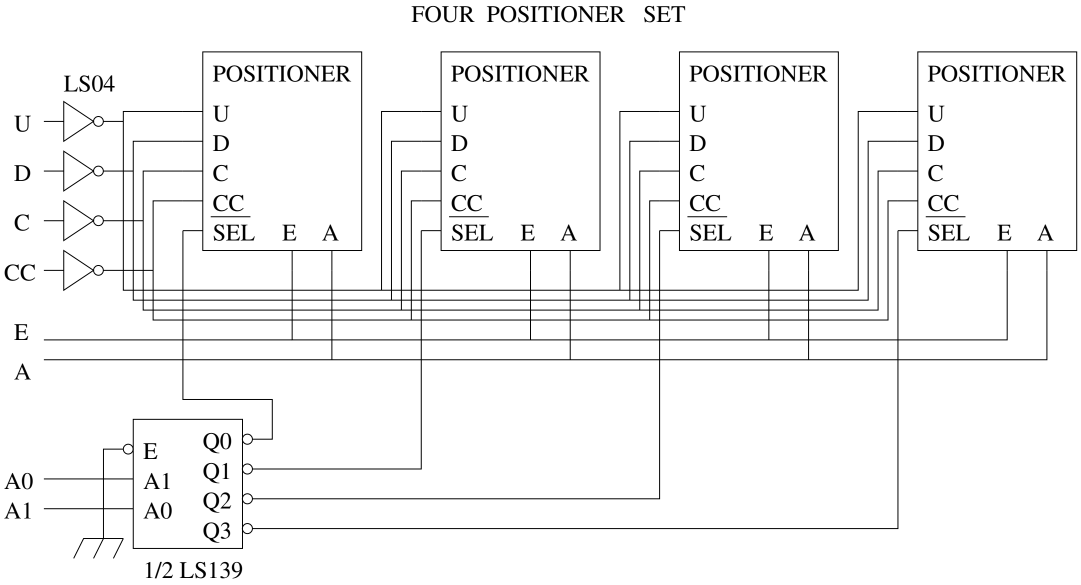

Figure 4: Schematic diagram of the control connections to the four antenna positioner circuits shown in Figure 5.

2 Positioner Control

Up to four, two-axis positioner control circuits may be attached to the microcontroller. The control and position readout lines are tied in parallel as shown in Figure 4, and the active controller is selected with two address bits, A0 and A1. The schematic diagram for one control circuit is shown at the top left of Figure 5. The Yaesu control box has four control pins on its remote control connector, one for each motion of up, down, clockwise, or counter-clockwise. The selected motion is activated by shorting the appropriate pin to the common pin on this connector. This is done with an HSR312 solid state relay chip as sown at the bottom right of Figure 5. The Yaesu control line requires about 8 milliamps of sinking current at +13 volts. In the parallel connection shown, the HSR312 is capable of sinking 320 milliamps with a resistance of 3 ohms. The HSR312 is an opto-isolator whose photodiode requires at least 3 milliamps of drive current with a forward voltage of 1.3 volts to turn the switch on. About 5 milliamps of current is supplied by a 2N3094 transistor to minimize the current required from the open-collector 74LS05 logic chip in the 'switch off' state.

The Yaesu control box remote control connector also contains two pins that supply voltages that are proportional to the angular position of the two axes. The voltage range is 0 to +5 volts with a supply impedance of about 250 ohms. Two of the HSR312 solid state relays are used to connect these monitor pins to the microcontroller A/D converters when the positioner is selected. In the 'off' state the HSR3123 leakage current is less than 1 microamp, which should not affect the accuracy of the selected positioner readout.

3 RF Transfer Switch Control

The circuit at the top right of Figure 5 shows the control connection to a Transco 700C70200 RF transmission line transfer latching relay. The switch is connected so that the filter is in the signal

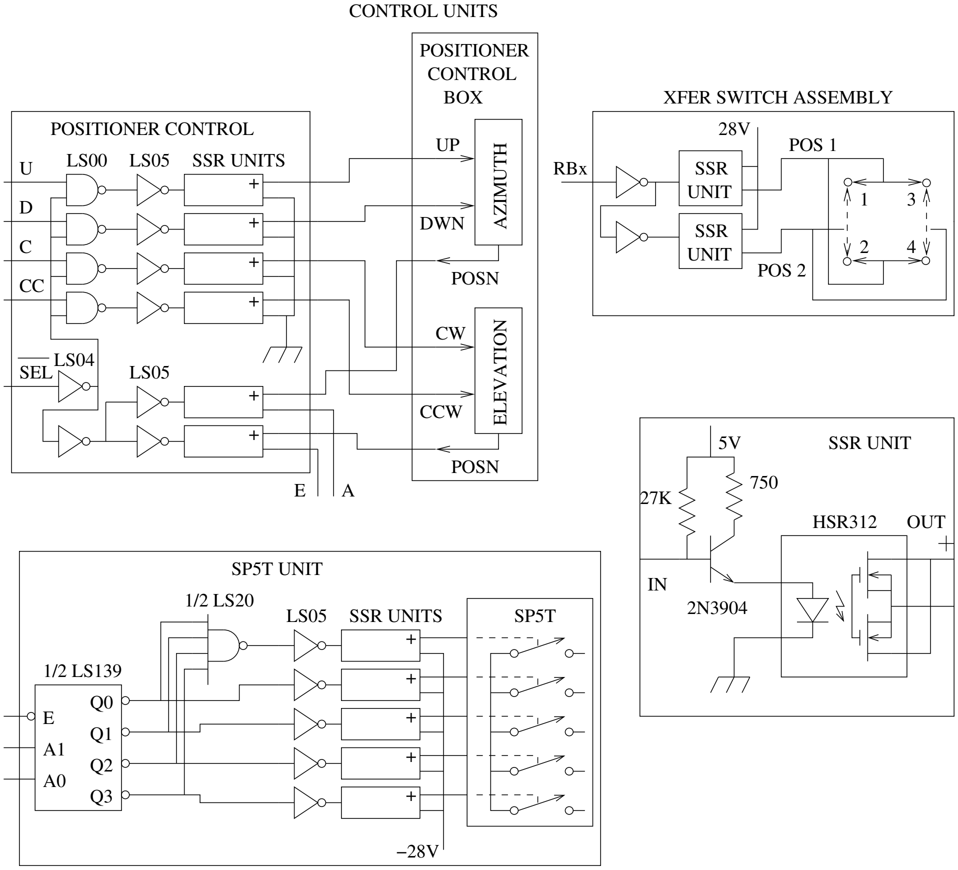

Figure 5: Schematic diagram of the individual control units shown in Figures 2 and 4.

path between ports 2 and 4. When the swtich is in position 1 the signal passes from port 1 to port 3, and the filter is shorted. In position 2 the signal passes from port 1 to port 2, through he filter to port 4, then back to port 3. Since the relay latches, no current is drawn except during the switching operation.

4 SP5T RF Switch Control

The control circuit for each of the two SP5T RF switches that select the antenna to be connected to the signal line to the lab is shown at th bottom left of Figure 5. These switches are Ducommun Technologies model 5SS2P12. Each switch position requires 135 milliamps of current at -28 volts to close. Only one position should be closed at a time. A 74LA139 decoder IC and a 74LS20 4-input NAND gate are used to activate one of the five switch relays according to the selected 3-bit address on the 74LS139 chip. The HSR312 solid state relay is used to supply current to the selected relay as described in Section 2.

5 Serial Port Command and Monitor Protocol

Commands to the PIC16F877A microcontroller are received by its serial port with be ASCII character packets terminated by a carriage return character (hex 0D). The available commands are given in Table 1. For example, to drive positioner 2 to an elevation where the A/D reading is decimal 856, first, select the positioner with the command P2, and then send the command E0856. The calibration of A/D readings in terms of azimuth and elevation in degrees will be the responsibility of the PC sending the commands. The full range of 10-bit A/D readings will be used since this range corresponds to the positioner read-out of about 0 to 5 volts. If a command outside of the available range is issued, the positioner will run to the limit of its motion in that direction and stop. A new position command will cancel any motion currently in progress from a previous command, if that commanded position had not been reached.

The PIC16F877A microcontroller will continuously send position data for the currently selected positioner to the PC approximately once per half second. Each command is echoed by the microcontroller. If an unknown commend is received, a 'U' message will be sent by the 877A. If two commands are received too quickly for the first one to be processed, a missed command, 'M', will be sent by the microcontroller. The message codes are given in Table 2.

6 Microcontroller Code Logic

The program for the PIC16F877A microcontroller is written in C and compiled with the development system from Custom Computer Services, Inc., http://www.ccsinfo.com/picc. This system runs on the MS Windows computer, LYCASTE. The resulting object code is loaded into the 877A via an in-circuit programming module (ICD-S40) from CCS that connects to the serial post of LYCASTE. See Section 7 for the details of running the compiler and in-circuit programmer. The C code is reasonably well documented so the details of the program can be found in the RFIstation.c file.

In this application, the PIC16F877A is programmed to initialize the default switch settings and turn all positioner motion off. It then enters an infinite loop watching for one of the three interrupts listed in Table 3. The 877A's internal TIMER1 continuously generates interrupts approximately once per 100 milliseconds. Each TIMER1 interrupt initiates a conversion by the analog-to-digital converter. When the conversion is complete the 877A receives an AD interrupt which causes the A/D data register to be read and stored. The A/D reads the azimuth and

Table 1: Monitor station command codes

Command	Action
Pn	Select the positioner (1-4) to receive subsequent move commands
Aaaaa	Drive the currently selected positioner to the azimuth where the A/D reading is the decimal number aaaa. aaaa can be between 0000 and 1023
Eeeee	Drive the currently selected positioner to the elevation where the A/D reading is the decimal number eeee. eeee can be between 0000 and 1023
Hx	Halt azimuth (x = A) or elevation (x = E) motion on the currently selected positioner
Fnx	Set filter switch pair n (1 to 5) to the state designated by x. (x = 0: filter in, x = 1: filter out)
Snx	Set the SP5T antenna selector switch number n (1 or 2) to position x (1 to 5)

Table 2: Monitor station message codes

Message	Meaning
U	Unknown message received
M	Missed message because two or more received too quickly
Paaa,eee	The current positioner A/D decimal readings are azimuth = aaa, elevation = eee

Table 3: Internal interrupts used in PIC16F877A program

Interrupt	Source
RDA	New character available in the serial port input register
TIMER1	Timer 1 internal counter has rolled over to 0 from its maximum value. This timer is set to run on the 877A's 10 MHz internal clock divided by 16 so that the timer's count increments every 1.6 microseconds and rolls over every 0.105 seconds
AD	Signals the completion of an A/D conversion ready to be retrieved

elevation encoders on alternate 100-ms interrupts. Every tenth TIMER1 interrupt the current azimuth and elevation values are sent to the host computer via the RS-232 serial link as an ASCII message in the format, Paaa,eee, where aaa and eee are the decimal readings from 0 to 1023 proportional to azimuth and elevation, respectively.

Each character received by the 877A via its serial input triggers an RDA interrupt, which, in turn, causes the serial port register to be read and stored in the incomming message buffer. When a carriage return character (hex 0D) is received the message is considered complete, and the new_msg_flag is set. The main program loop will see this flag go up and begin parsing the message and acting on it according to its contents as listed in Table 1. If the command is for motion of one of the positioner axes, the motor will be tuned on in the appropriate direction. The main program loop will then watch the encoder readings and turn off the motor when the commanded position is reached.

7 Running the Compiler and In-Circuit Programmer

To start the CCS development system log onto the MS Windows computer, LYCASTE with

User: rfisher

Password: Ki1k∧b

Then open the command prompt window, change directories to "\Program Files\PICC", and run pcw.exe. This will open the CCS development environment. Now open the RFIstation project through the menu selections Project - Open. This should display the C program in the editor.

To compile the program press F9 or use the Compile menu selection or click on the compile toolbar icon (a little yellow file folder with 1010 below it). The object code may then be loaded into the 877A. First, turn off power to the microcontroller board. Connect the 6-pin modular plug to the ICD-S40. Turn back on the power to the microcontroller board. Select the Tools menu, and select ICD. This will start the download, which will take 20 or 30 seconds. After this completes turn the power to the microcontroller board off. Disconect the 6-pin modular plug from the ICD-40. Finally, restore the power to the microcontroller board, which will start the 877A program.

8 Communicating with the Microcontroller from a Linux Computer

The RFI station microcontroller may be controlled and monitored from any computer with an RS-232 serial port. The port setup is 9600 baud, 8 bits, one stop bit, no parity, no RTS/CTS flow control.

In the Linux directory /users/rfisher/Applications/RFImonit/MonitStaCtrl there are two programs, 'send' and 'receive', that can be used to communicate with the microcontroller. The C/C++ code for these are in the files send.cc and receive.cc.

First, connect one of the Linux computer serial ports to the microcontroller board's serial port. Next, open two xterm windows and cd to /users/rfisher/Applications/RFImonit/MonitStaCtrl in both. In one window start the receive program with 'receive port#'. For example, if you are using com port 1 on the Linux computer, use

receive 1

You should see "Ready to receive" printed followed by a position readout message every second, if the microcontroller is running. In the other xterm window start the send program with 'send port#', e.g.

send 1

This will give you a ' > ' prompt. At this prompt type any of the commands in Table 1 followed by Enter. To terminate either the 'send' or 'receive' program hit Crl-C in its xterm window.