Main PIC Programming Guide

Memory

Program memory - 4 slots
- Main program in slot 0
- Program in slot 2 for retrieving data and resetting everything
- Program in slot 3 for testing everything on the board

Main variables – 56 bytes
- (Assignment later)

PIC RAM – 256 bytes
- The first 56 bytes are b0-b56, but the rest can be accessed with peek/poke. Currently unassigned, but will probably be used as a buffer for SPI commands

Scratchpad – 1024 bytes
- Most of the time serves as the input buffer for the radio
- Also used for output buffer when sending
- Input buffer when getting data from phone (perhaps)

Table ROM – 256 bytes
- Used to store the ROM parts of the output string
- Also for command and password strings (0x80 onwards)

RTC ram – 256 bytes
- Used to save sensor data temporarily before being sent

PIC NV data
- Used to store data such as the pointer for the flash memory

External flash – 4Mb (2048 pages, 256 bytes per page, in 8 sectors)
- All data sent is also logged on the flash
- One 256byte page is used per reading, giving 2048 readings

NV variables

- **Next** page/packet pointer (0x50/1)
- Highest altitude (so the payload can work out if is on its way down) (0x52/3)

Volatile variables

- Descending hysteresis counter (used by the phone)
- Current altitude (so no need to keep asking the GPS – this variable is read a few times
- RTC RAM pointer

PIC Program Cycle

Check GPS position
    Add to RTC RAM
    Check altitude/descending -> send text?
Take picture
Check each sensor
    Add to RTC RAM
Check radio RX buffer
    Process any commands
    Add response to RTC RAM
Turn off RX, disable hserin
Check for texts (may require scratchpad)
Copy RTC RAM data to scratchpad
Copy scratchpad data to external flash
Transmit scratchpad data
Enable RX/hserin

Other Program Aspects

'Kill timer' – every cycle the PIC resets the timer, but if it reaches, say 2mins, something's gone wrong, and the timer resets the PIC

Update the RTC with the GPS – the RTC can then be used when the GPS has no lock

At the start of the main program, it waits for 2secs for a 'C' from serrxd, which will cause it to switch to slot 2 which has the control panel code

Main variable (b0-b55) assignment

Subject to change

b0 – b9 Result variables – used by functions such as GetTime to leave the result of the subroutine in memory for the calling code to use. The meaning of each variable depends on each function

b10-b14 Input variables – used to send data to functions, such as what channel the ADC should read

b15 – b39 Free use, but should expect the contents to be changed outside a small bit of code. If data is needed to exist for longer, put it in the global variables

b38:b39 (w19) Next packet ptr – retrieved at the start of each cycle from the PIC NV data

b40-b49 Low level routines (can be used as temporary variables for a short length of time if any of the routines that use these variables are not called – probably best to avoid this as there should be enough space)
b45-b49 - Used by RTC, flash, ADC
b40-b45 – Used by slightly higher level commands

b50-b55 Global variables – do not use for other reason then stated
b50:b51 (w25) – Current altitude
b52 - Descending hysteresis counter
b53 - RTC RAM pointer
b54 – Table pointer
b55-b55 unassigned

Table Assignment

The table will be used to store the longer ROM parts of the string, in a format which allows variable length for each part of the string.

The string will be in the following format, where each field is separated by a comma

<string start>, MORE

The typical APEX string will be

$$APEX

Commands

FlashSingleOpCode:
b10 – opcode

-Writes a single opcode to the flash only

FlashRead_Byte:
b10 - opcode
b0 - byte value

used to read a byte value after sending the OPCODE (eg status register) NOT for reading a byte from memory

FlashWrite_Byte:
b10 - opcode
b11 - byte value

used to write a byte value after sending the OPCODE (eg status register) NOT for writing a byte from memory

FlashReadSpad:
main function for reading data into spad
w5 (b10:b11) - bytes to read

b13 - addr start LSB		byte in page
b14 - addr start	(w7)	page select LSB
b15 - addr start MSB	(w7)	page select MSB

b13:b15 select the address, b14:b15 (w7) select the page, and b13 the byte on the page

FlashWritePage:
main function for writing data from spad
b10 - end addr in spad (count zero based)

b13 - addr start LSB		byte in page
b14 - addr start	(w7)	page select LSB
b15 - addr start MSB	(w7)	page select MSB

b13:b15 select the address, b14:b15 (w7) select the page, and b13 the byte on the page

RTCReadSingle:
b10 - address
b0 - read value

RTCWriteSingle:
b10 - address
b11 – value

RTCRAMWriteSingle:
b10 - RAM addr
b11 - RAM data

RTCRAMReadSingle:
b10 - RAM addr
b0 – value

RTCRAMClear:

RTCRAMReadSpad:
b10 - start address
b11 - end address
data saved in spad

RTCRAMWriteMany:
writes a number of bytes to ram from the PIC RAM (remember 0-56 of ram = b0-b56)
b10 - RTCRAM start addr
b11 - PIC RAM start addr
b12 - PIC RAM end addr

WriteComma:
Increments RTC RAM ptr

bcd_decimal:
input/output on b10,b11,b12,b13

ADCshift:
b10 selects channel - remember selects channel of NEXT reading
w0 is output word

GetLatitude:
b0 - X if timeout
result in variables b1-b9
b9 - E/W or ',' for no fix

GetLongitude:
b0 - X if timeout
result in variables b1-b10

GetUTCTime:
b0 - X if timeout
b1,b2 - hour
b3 - ':'
b4,b5 - minute
b6 - ':'
b7,b8 - sec