Help sheet 20

ST Robotics

Data logging.

There are many ways of doing this. Assuming you have some data input, for example from the ADC on the multi-IO card, you can save a reading to RAM then increment the RAM for the next reading.

In the following example we are collecting data in 2 bytes of memory per reading. We then need to increment memory by 2. In this example we have chosen an area of unused RAM from 2000 to 3000 but if you need more then pick something near the top of low memory, beyond the likely top of your programming (it would be a very large program indeed that went past C000)

Initialize with INITRAM
Get the data on the stack and follow with SAVEIT

```
HEX
2000 CONSTANT LRAM ( THIS IS IN LOWER MEMORY WHERE THE BINARY DATA WILL GO
USER RAMPTR ( THIS IS A POINTER TO WHERE THE DATA WILL GO
: INITRAM
0 LRAM LRAM 0FFF + FILL ( CLEAR LRAM
LRAM RAMPTR !
;
: SAVEIT
RAMPTR @ !
2 RAMPTR +!
RAMPTR @ 2FA0 > IF -2 RAMPTR +! ." MEMORY FULL " QUIT THEN
;
```

When you have enough data (or memory is full) then you can upload to the PC using Robwin. Click file, save binary, bank 0, start 2000, length 1000 (or less but always round up to the next 100 hex block)

Next you can create a csv file.

```
HEX
8000 CONSTANT FILE ( THIS IS WHERE THE CSV FILE WILL BE ASSEMBLED
                   ( IN *UPPER* RAM
USER FILEPTR       ( THIS IS A POINTER TO THE START OF THE FILE
: INITFILE         ( FILLS FILE WITH SPACES
0 ( TOP OF RAM ) FILE DO
   ASPACE I E!
LOOP
FILE FILEPTR !
;
USER RDGS
: MAKEFILE
INITFILE
RAMPTR @ LRAM - 2 / RDGS !
." readings " RDGS ? ." OK? "
ASK DUP 0 > IF RDGS ! ELSE DROP THEN
RDGS @ 2 * CR 0 DO
  I LRAM + @ #.
  ASPACE PAD E!
  PAD 1+ FILEPTR @ 7 MOVUP
  2C FILEPTR @ 7 + E!
  8 FILEPTR +!
2 +LOOP
CR
1 . FILE X. FILEPTR @ FILE - FF00 AND 100 + X.
;

DECIMAL
```

When the data collection is complete enter MAKEFILE
This replies with the number of readings collected. To use less than that simply enter a lower number.

MAKEFILE does fill the screen with numbers, ignore them. The file itself is assembled at FILE (8000 in our example) in memory bank 1. As it works it creats a series of 8-byte (8-charater) fields. Each number is in a 7 character field with leading spaces. The 8th character is a comma.

When finished the save file parameters are shown on screen. Save the file to PC using those numbers – bank 1, start 8000, length as shown.