**Using 2D Look-Up table to convert to ASCII**

Data from UART1 is converted to ASCII format and written thru UART2 to a Logomatic V2 microSD data logger.

The source is Microstrain 3DM-GX1 set up to output data in 11-byte sets wit header byte0x0E. The look-up table is organized in rows. Each row contains the numerical values corresponding to the ASCII code for three digits including any leading zeros followed by 44 (comma) and 32 (space). The input to the look-up block is a number used to select the column. The table is zero-indexed; the first column is 48 48 48 44 32 corresponding to “000,[space].” The table has 256 columns to handle 8-bit integers.

[48 48 48 48 48 48 48 48 48 48...,

48 48 48 48 48 48 48 48 48 48...,

48 49 50 51 52 53 54 55 56 57...,

44 44 44 44 44 44 44 44 44 44...,

32 32 32 32 32 32 32 32 32 32...]

The table is generated with a Matlab file:

T = ones(5,256);

T(4,:) = 44;

T(5,:) = 32;

for k=0:255

T(3,k+1) = mod(k,10);

T(2,k+1) = (mod(k,100)-T(3,k+1))/10;

T(1,k+1) = floor(k/100);

end

for k=1:256

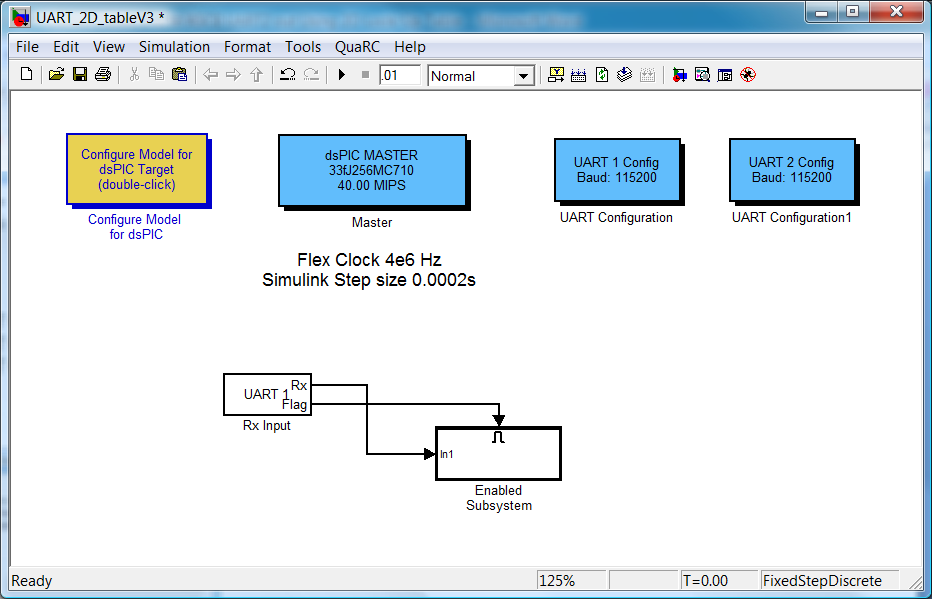
T(3,k) = T(3,k) +48;

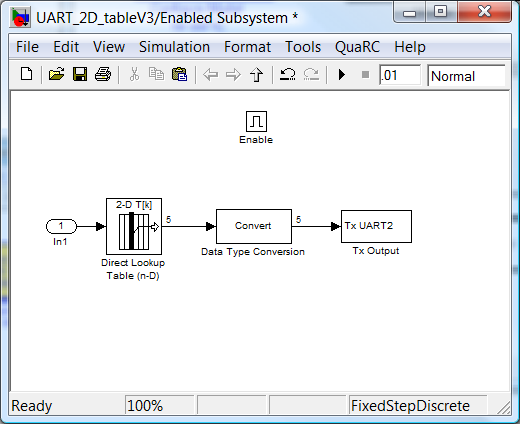
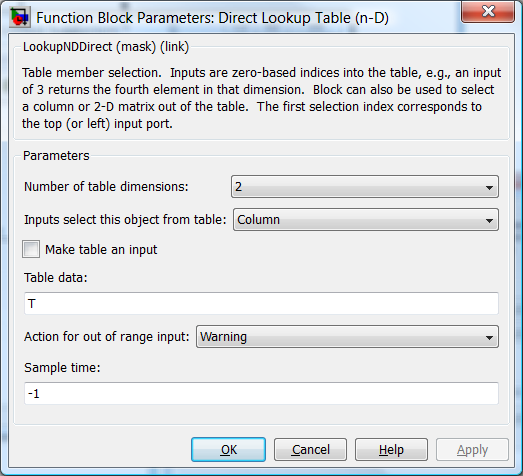
T(2,k) = T(2,k) +48;

T(1,k) = T(1,k) +48;

end

The Simulink model is triggered based on UART1 Rx Flag.



Procedure:

Microcontroller must be on but NOT outputting data on UART to begin, (e.g Erase Device in MPLAB)

SD card logger must be connected to microcontroller UART and powered on after microcontroller power, lights flash alternately and go off

Then program is run and data is recorded.

Sample Data from the sd card, I added the highlight of the header byte:

201, 158, 149, 111, 014, 255, 054, 255, 019, 164, 124, 242, 160, 149, 115, 014, 255, 054, 255, 019, 164, 124, 242, 162, 149, 117, 014, 255, 054, 255, 019, 164, 126, 242, 164, 149, 121, 014, 255, 054, 255, 019, 164, 126, 242, 166, 149, 123, 014, 255, 055, 255, 019, 164, 126, 242, 168, 149, 126, 014, 255, 055, 255, 019, 164, 128, 242, 170, 149, 130, 014, 255, 056, 255, 021, 164, 129, 242, 172, 149, 136, 014, 255, 056, 255, 021, 164, 129, 242, 174, 149, 138, 014, 255, 055, 255, 021, 164, 129, 242, 176, 149, 139, 014, 255, 055, 255, 019, 164, 126, 242, 178, 149, 136, 014, 255, 056, 255, 021, 164, 128, 242, 180, 149, 143, 014, 255, 056, 255, 021, 164, 128, 242, 182, 149, 145, 014, 255, 058, 255, 021, 164, 126, 242, 184, 149, 147, 014, 255, 058, 255, 021, 164, 126, 242, 186, 149, 149, 014, 255, 058, 255, 021, 164, 126, 242, 188, 149, 151, 014, 255, 058, 255, 022, 164, 126, 242, 190, 149, 154, 014, 255, 056, 255, 023, 164, 128, 242, 192, 149, 157, 014, 255, 058, 255, 023, 164, 128, 242, 194, 149, 161, 014, 255, 058, 255, 024, 164, 128, 242, 196, 149, 164, 014, 255, 058, 255, 024, 164, 130, 242, 198, 149, 168, 014, 255, 059, 255, 024, 164, 131, 242, 200, 149, 172, 014, 255, 059, 255, 024, 164, 132, 242, 202, 149, 175, 014, 255, 059, 255, 024, 164, 133, 242, 204, 149, 178, 014, 255, 059, 255, 024, 164, 132, 242, 206, 149, 179, 014, 255, 059, 255, 024, 164, 132, 242, 208, 149, 181, 014, 255, 059, 255, 026, 164, 131, 242, 210, 149, 184, 014, 255, 059, 255, , 106, 154, 125, 01

Matlab will be used to parse.