CPTR-215
HW#20

Design and implement an assembly language program that will “drive” the class vehicle autonomously to follow a black line on a white piece of paper.

Pressing the *center* joystick switch will cause the vehicle to stop. Pressing joystick *up* will start vehicle motion forward. Any other functionality of the joystick can be defined by you. There is a switch on P0.14 that you may also use if a switch other than the joystick is an advantage for you (see page 18 & 19 of the User Guide for the Embedded Artists Board).

Design your program before coding it. Use a flow chart, NS diagram, state diagram, or a combination thereof to document your design. These can be drawn neatly by hand. You don’t need to spend time electronically drawing them.

To turn in:

1) Hard copy of your .s file annotated with your comments about if or not the program works and any comments about challenges you overcame to make it work.
2) Hard copy of your design document(s)
3) Email me a copy of your .s file.

Notes:

Vehicle sensor configuration

There are two infrared sensors, each containing an integral infrared light source, that are mounted on the front of the vehicle pointed downward. As the vehicle encounters changing materials or colors under it there are changes in reflected light intensity which create a changing voltage. The sensor output voltage is connected to an analog to digital converter (A/D) in the LPC-2148 microcontroller chip. The A/D converts the voltage to a binary number which can be read and used in your program. The angle of the light source vs sensor causes maximum output from the sensor when it is about 0.15 inches from a surface. The circuit on the vehicle is arranged to create approximately 0.5v when it sits on a white piece of paper. If black paper or a black line is under a sensor the voltage is considerably lower.

The two sensors are mounted side-by-side with a small space between. If the vehicle is placed on a white paper centered over a bold black line that has been drawn on the paper, both sensors should produce about the same voltage. If the car moves so that the left sensor is over the black line but the right is over white paper, the right will have much greater voltage output. A control program would then slow the left wheel to bring the vehicles trajectory back to a more centered-on-the-line position. If the car strayed left the opposite situation and correction would occur.

A/D subsystem details

There are two 10-bit A/D converter circuits (AD0 and AD1) in the LPC-2148 each with a seven-input multiplexer circuit ahead of it to select one of seven voltages to be connected for measuring. (A multiplexer is just a switch that can be controlled electrically). As with the pulse width modulator, there are a series of I/O addresses that are used to configure and control operation of the A/D subsystem. A 10-bit converter creates a 10 bit long data word. In the LPC-2148 only positive voltages can be measured and thus an unsigned binary value is produced.

To demonstrate A/D operation on the Embedded Artists board, there are two variable resistors (just left of the LEDs) with red knobs. These are configured as voltage dividers. Changing their position changes an output voltage that is proportional to their present position.

A/D subroutines for you to use

I have created a demonstration program named demo_ad.s that reads the left voltage divider on port P0.28 and displays the upper 8 bits (out of 10) on the 8 LEDs (turn the left red knob and the ARM board to vary the voltage).

Also, A/D subroutines named ad_init and ad_read are available for you to use in HW20 to read A/D channels AD1.6 and AD1.7 to which microcontroller pins P0.21 and P0.22 are connected. P0.21 is connected to the left sensor and P0.22 to the right sensor on the vehicle. See the headers of the subroutines for more info about their use. There is also a demo program named demo_ad_subs.s that shows these subroutines in use.

The full scale input voltage for the A/D is about 2 volts and thus the largest data value returned by the A/D will be 3FF (hex). The expected range of values returned from the A/D when reading the light sensors on the vehicle will be 0 to about 0.5 volts or 0 to about FF (hex). When data is read from the A/D output register the 10 bits of interest are in bit locations 15 down to 6. The ad_read routine shifts it over to be right justified.