Homework 4-Follow the Light

Project Statement

In this project, you will use a mechanical servo mounted with a Light-Sensitive resistor to mechanically respond to a moving light.

Deliverables:

- All code you use to implement the program;
- A report with the following
  - A clear statement on the completeness of your system. Report what testing procedures you performed.
  - Document design decisions
  - Provide a usage manual (make it clear what your system DOES not what it was intended to do but you never got working).

Grading:

The grade distribution for this project will be as follows

- 15% Coding style
- 20% Report
- 65% Functionality

The functionality will be broken down into components. Each working component will give you credit. A fully working system will give you full credit for this portion.

Description:

The servo's movement range is about 180 degrees and is controlled with a PWM signal. The signal's frequency should be around 125 Hz and the pulse width should correspond to approximately 6% to 30% duty cycle (experiment a little to find what works best for you). The servo moves the arm in the range according to the width of the pulse.

The system is primarily defined by two behaviors FULLSWEEP, LOCALSWEEP and two modes, FOLLOW THE LIGHT and AVOID THE LIGHT.

Modes:

Follow the Light Mode - Objective is to find the angle of **MOST** illumination. The LCD should display 'FTL' and pressing UP on the joystick activates this mode.

Avoid the Light Mode - Objective is to find the angle of **LEAST** illumination. The LCD should display 'ATL' and pressing DOWN on the joystick activates this mode.

Behaviors:

In each mode, there are two behaviors: **FULLSWEEP** and **LOCALSWEEP**. The **FULLSWEEP** behavior is automatically followed by the **LOCALSWEEP** behavior. While in the **FULLSWEEP** behavior, the system should not respond to any button presses.

FULLSWEEP: A full sweep of the range should be performed starting at 0 ° and sweeping to 180 ° in about 20° increments. An illumination measurement should be taken at each step. The optimum angle (most light or least light depending on the mode) is decided, called the initial ***primary angle*** and is displayed on the left 3 character positions on the LCD as a value in the range [0,180]. The next behavior should then commence.

LOCALSWEEP. Starting at the ***primary angle,*** the servo should continuously move in the cycle given below, finding the optimum of the three angles.

primary angle* □ *(primary angle - 10 °) □ (***primary angle*** + 10 °)

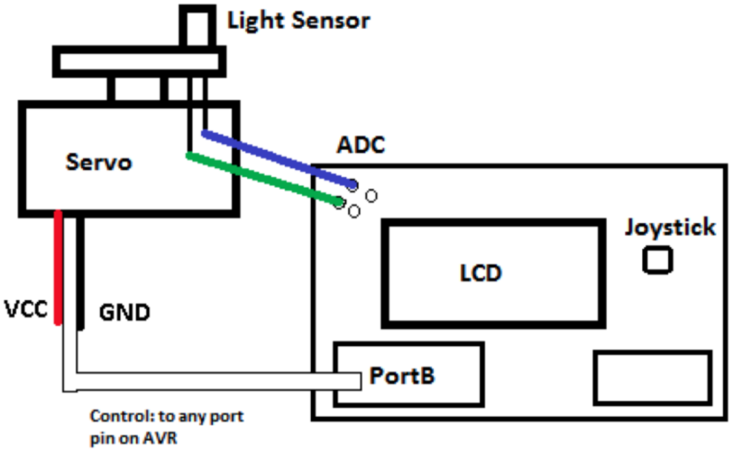
After each cycle, a new ***primary angle*** should be decided and displayed. If the ***primary angle*** is at the limit of the servo range, the local search pattern will not involve one of the +10°or -10° measurements.

Further Requirements and Instructions:

Up and Down and Right button presses must be detected with Pin Change Interrupts.

Floating point operations are not permitted (unless you make a case you need it).
The servo control PWM signal must be driven by the Timer 0 waveform generator hardware on PB4.

System Block DIagram:



Light Dependent Resistor(LDR):

The light measurement is done by mounting the light dependent resistor (LDR) provided to you, to the servo (with tape) and connecting wires from it to your board. The LDR measurement will be taken using the ADC on the AVR and there are two ways you can connect it to the butterfly, by using ADC2 **or** ADC1. Both methods are covered below **but you only need to use one of them.**

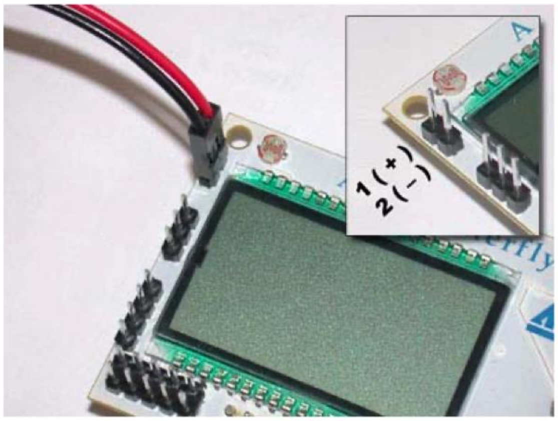
Using ADC2 (Channel 2 on the ADC): (need to double check this)

This is the connection suggested by the butterfly documentation and is the one shown in the block diagram above.

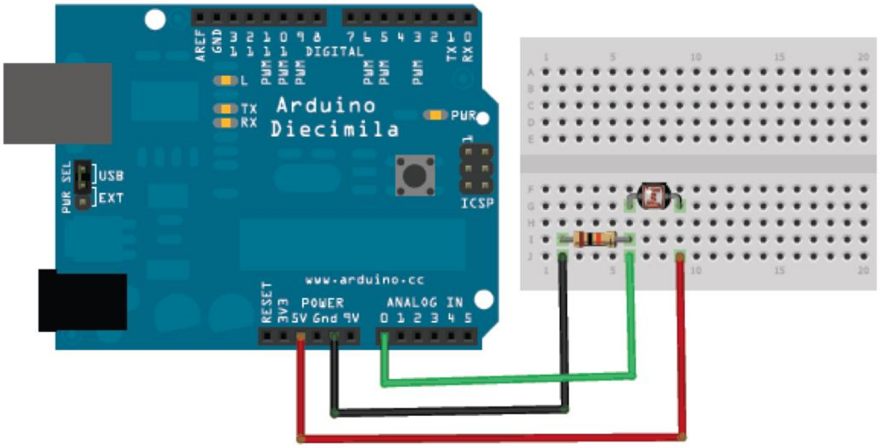
Once connected in this format, you would take the reading from the ADC1 channel.

Using ADC1(Channel 1 on the ADC):

Alternatively you can connect the LDR to ADC1 which is located on the left hand side of the LCD. See image below for location.



One side of the LDR would need to be connected to VCC and the other side would go to ground (the V- on the butterfly) through a pull down resistor (10K) with the analog signal coming from here to the AVR. See images below for connection examples from the data sheet(**Note** in the example below, the AVR Butterfly would take the place of the Arduino and you can use any VCC connection on the butterfly for the red wire)..



You can find more information on the LDR in the datasheet at the link below:

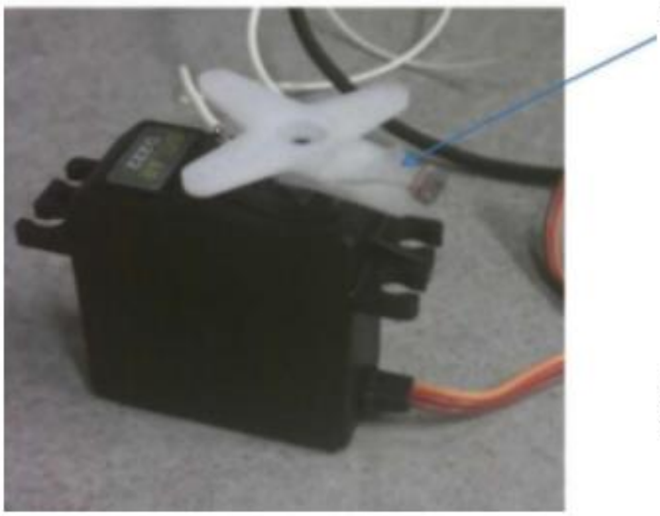
LDR DataSheet

Most of the kits do not have LDRs so you should plan to pick these up as soon as they are available. More information will be shared regarding when and where you can pick these up.

Servo:

The servo will be connected to AVR with the LDR taped to it. The black wire on the servo should be connected to ground (the same ground that everything else is connected to), the red to a high voltage (4.8-6V) and the white is the PWM signal coming from the AVR.

Power Supplies will be provided in ITE 375 if needed.



This is the light sensor, simply taped on, with wire wires going to the Butterfly

black : ground, **red** : 4.8 to 6V,
white : PWM signal from

Code Libraries:

You can use the code libraries for the ADC and LCD display provided during discussion but you will need to edit them to fit your needs for this project. You will also need to run some tests to see how the LDR works with the ADC and how the values created correspond to the intensity of the light it is exposed to.

You need to keep in mind that this project involves software and hardware integration so you should test your design as you go. Sometimes the issue is the code and sometimes it is the hardware and sometimes it is with the integration so you shouldn't wait to test until integrating everything. You can use oscilloscopes and multimeters in the lab to test your hardware. You can also use the UART on the AVR and RealTerm as a way to test certain values. Finally, Atmel Studio also has a debug mode that you can use as well.

Most importantly, do NOT procrastinate!