B+

UNIVERSITY OF CALIFORNIA, IRVINE

Electrical Engineering and Computer Science Department

AUTOMATED WINDOW SHADE

EECS 129B
Professor:
Raymond Klefstad

Group members:
Thinh vu (
Loc tai (
Hyun Shin

TABLE OF CONTENT

LIST OF ILLUSTRATION

Abstract:

We set out to create a project that would incorporate the knowledge attained last quarter using a photocell, micro controller and the necessary circuitry. The project itself was an idea that could extend to some meaningful and useful ways in which the normal shutter could be utilized. The two primary functions that were achieved by the automated shutter were the automated alarm clock and the privacy feature. The shutter would open in the morning causing the user to wake up naturally due to sunlight and on the other side of the spectrum shield against prying eyes during the night period when sunlight was no longer necessary. The program also would also allow users to have a third mode operation that would be at a 45 degree angle in times where the a smaller amount of light was necessary.

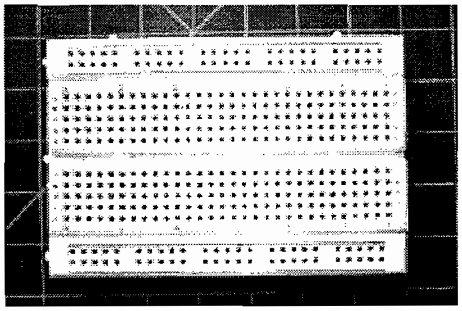


Figure 1: Solderless breadboard

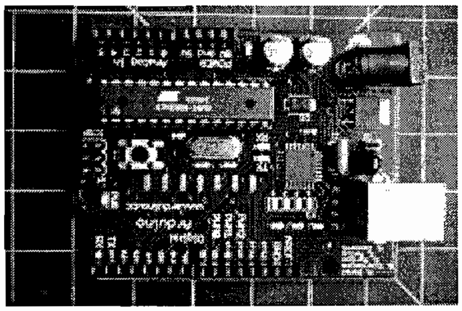


Figure 2: Arduino Microcontroller

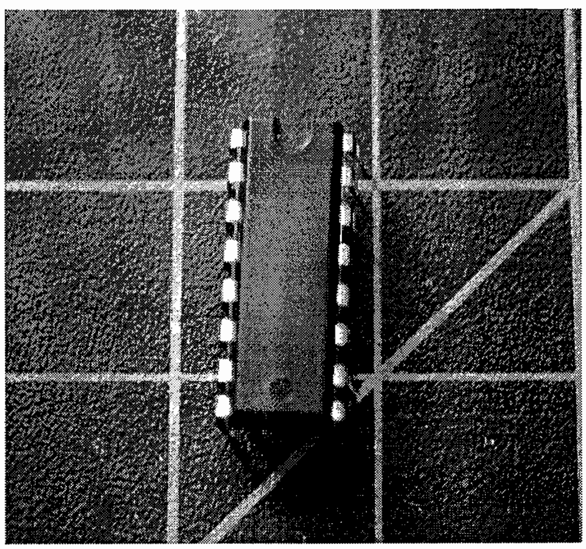


Figure 3: IC-HB1 H-Bridge

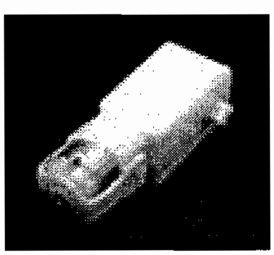


Figure 4: 5v DC motor

2) Project Description

The project that we designed this quarter is an automated window shutter system that takes into account the amount of sunlight on the outside and compares it to the amount of light on the inside to create an ideal lighting condition for the convenience of the user.

Purpose

When our group members were thinking about what kind of project to do, we realized that it would be a good idea if in the morning, instead of a habitual and annoying ringing noise that we hear from our alarm clocks, we could actually have the sun effectively fulfill the same role, waking us up naturally as sunlight did for the native Americans and past civilizations. Along with this idea, we thought that it would be a good idea for the shutters also to protect our privacy after a certain point in the day when it was no longer necessary to keep the shutters open.

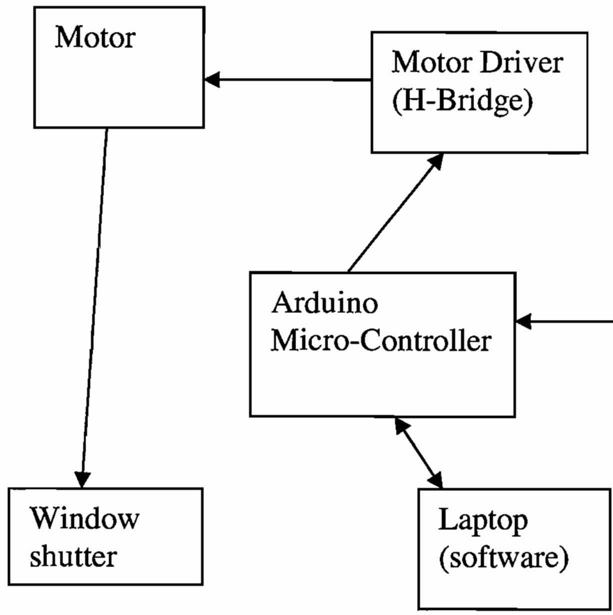
Background

We chose this project as a group because it seemed like a relatively simple way to create a project that we could enjoy working on. We also studied the mechanisms and functionalities of photocells and motors last quarter in the first of these senior design classes, and felt that it would be fun to implement these two devices into a project that would be useful and practical. As college students, we realized that waking up in the morning after studying late or partying late was one of the most difficult things to do, and to have an automated system that would help us completely to wake up or just to supplement our alarm clock was a worthy idea to research and to create our project on.

Features

As previously mentioned, our project has two main purposes. These purposes are to act as an automated alarm clock in the morning and also to protect our privacies at night when the shutter is no longer necessary to provide light.

3) System level Block diagram:

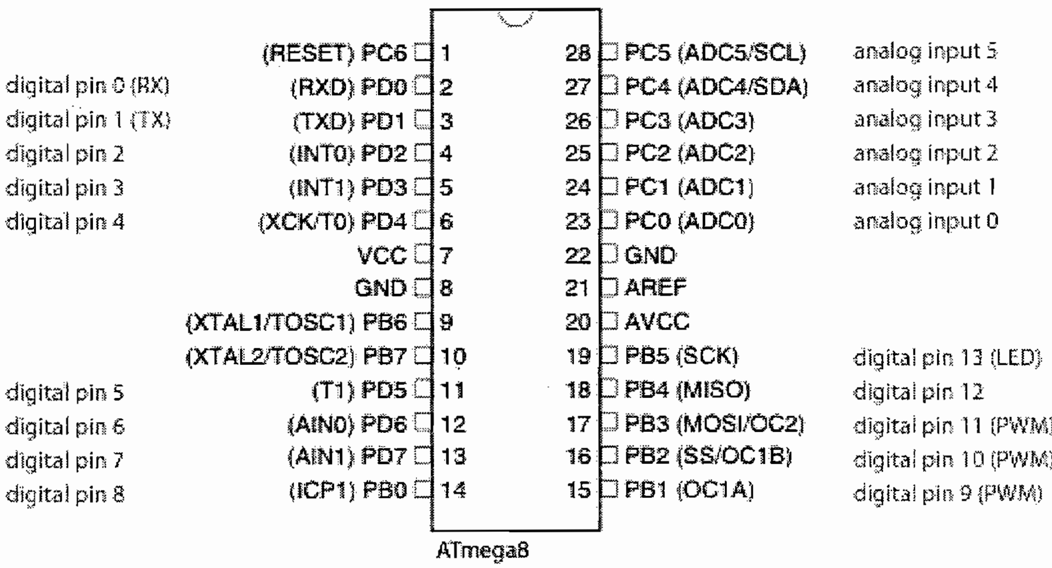


-laptop upload software into atmega8 u-controller on Arduino
-photocell provides amount lights as input to the micro-controller
-micro-controller computes and determines the result and sends it to H-Bridge chip
-H-bridge use the output from u-controller to drive the motor
-the motor will turns clockwise or counterclockwise accordingly
-shutter can be close, partly open, or completely open.

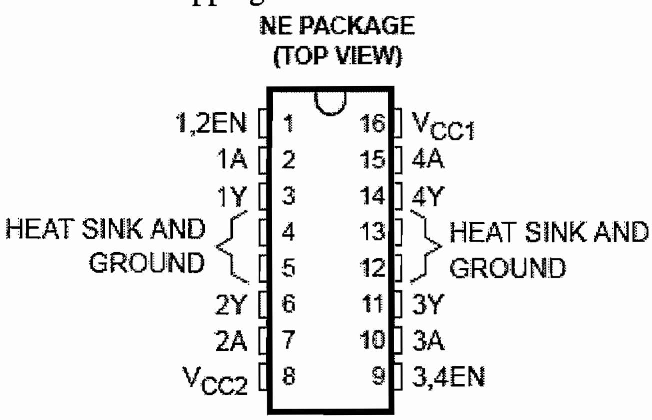
4) Circuit Level Block Diagram

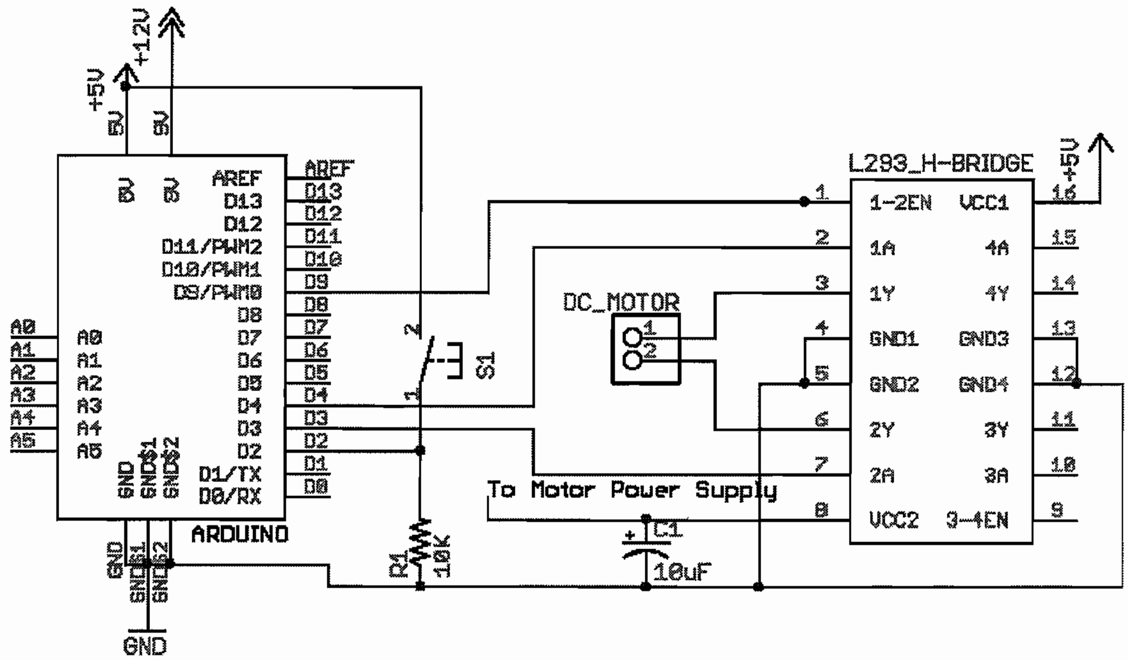
Arduino Pin Mapping

www.arduino.cc



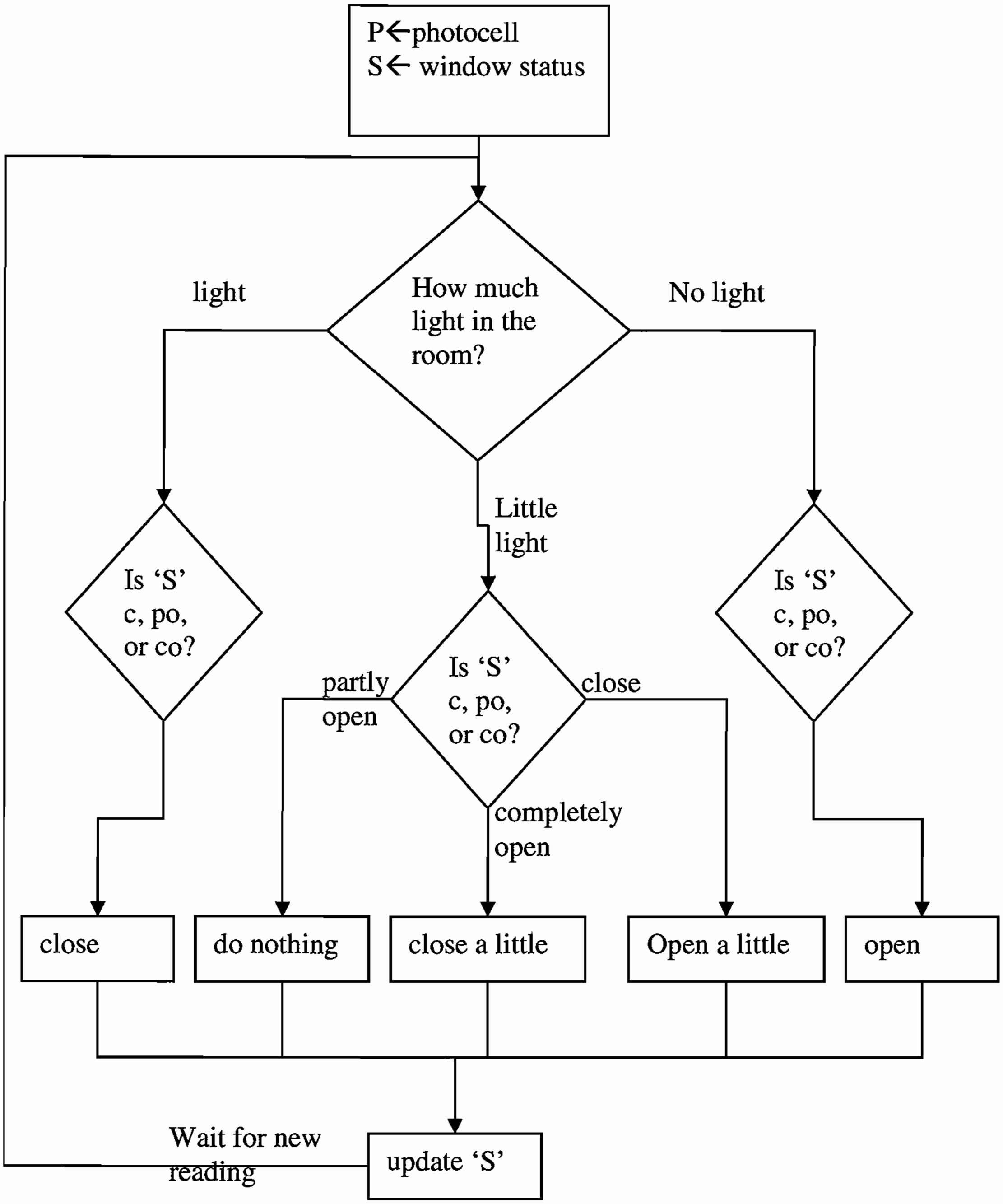
IC-HB1 PIN mapping





In this circuit diagram S1 represents photo sensor. Here the photocell acts as a switch to supply to the motor.

5) Software Descriptions



6) System Test Plan

The intension of this project is to create a simple and user friendly application. The photo sensor is located inside the box as a person is inside a room. Before apply power to Arduino board the shutter should be close and the top should be open allowing light to come inside the box. To test the project connect all the modules, upload the software to the microcontroller: now try to vary the amount of light inside the box, the shuttle should open and close accordingly.

7) Cost Analysis:

List of Components and cost

quantity --	description --	price
1	**Arduino USB board**	**$31.95**
1	**H-bridge Motor Driver**	**$3.90**
1	breadboard	$15.95
1	100' reel 22awg, solid black wire	$5.49
1	100' reel 22awg, solid red wire	$5.49
1	100' reel 22awg, solid yellow wire	$5.49
1	USB, a to b, cable	$1.55
1	10k resistors	$1.00
1	MOTOR,DC,10.3Krpm@4.5VDC	$16.15
1	PHOTOCELL	$2.12
1	10 uF Capacitor	$0.99
2	3/32x2x36 balsa	$1.63each
2	.01x1/2"x12" Stainless steel strip	$0.59each
1	Proxy glue	$4.69
2	3/32x36 music wire	$2.29
1 bag	Screws & nuts	$1.25
2	3' pine square sticks	$2.39each

Component Specifications:

Arduino Microcontroller

ATmega8 by ATMEL

High-performance, Low-power AVR® 8-bit Microcontroller
• Advanced RISC Architecture
– 130 Powerful Instructions – Most Single-clock Cycle Execution
– 32 x 8 General Purpose Working Registers
– Fully Static Operation
– Up to 16 MIPS Throughput at 16 MHz
– On-chip 2-cycle Multiplier
• Nonvolatile Program and Data Memories
– 8K Bytes of In-System Self-Programmable Flash
Endurance: 10,000 Write/Erase Cycles
– Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
– 512 Bytes EEPROM
Endurance: 100,000 Write/Erase Cycles
– 1K Byte Internal SRAM
– Programming Lock for Software Security
• Peripheral Features
– Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
– Real Time Counter with Separate Oscillator
– Three PWM Channels
– 8-channel ADC in TQFP and MLF package
Eight Channels 10-bit Accuracy
– 6-channel ADC in PDIP package
Eight Channels 10-bit Accuracy
– Byte-oriented Two-wire Serial Interface
– Programmable Serial USART
– Master/Slave SPI Serial Interface
– Programmable Watchdog Timer with Separate On-chip Oscillator
– On-chip Analog Comparator
• Special Microcontroller Features
– Power-on Reset and Programmable Brown-out Detection
– Internal Calibrated RC Oscillator
– External and Internal Interrupt Sources
– Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
• I/O and Packages
– 23 Programmable I/O Lines
– 28-lead PDIP, 32-lead TQFP, and 32-pad MLF

• Operating Voltages
– 2.7 - 5.5V (ATmega8L)
– 4.5 - 5.5V (ATmega8)
• Speed Grades
– 0 - 8 MHz (ATmega8L)
– 0 - 16 MHz (ATmega8)
• Power Consumption at 4 Mhz, 3V, 25°C
– Active: 3.6 mA
– Idle Mode: 1.0 mA
– Power-down Mode: 0.5 µA

IC-HB1 Motor driver: SN754410

QUADRUPLE HALF-H DRIVER by Texas Instruments
• 1-A Output-Current Capability Per Driver
• Applications Include Half-H and Full-H Solenoid Drivers and Motor Drivers
• Designed for Positive-Supply Applications
• Wide Supply-Voltage Range of 4.5 V to 36 V
• TTL- and CMOS-Compatible High-Impedance Diode-Clamped Inputs
• Separate Input-Logic Supply
• Thermal Shutdown
• Internal ESD Protection
• Input Hysteresis Improves Noise Immunity
• 3-State Outputs
• Minimized Power Dissipation
• Sink/Source Interlock Circuitry Prevents Simultaneous Conduction
• No Output Glitch During Power Up or Power Down
• Improved Functional Replacement for the SGS L293

DC Motor

This motor offers 50 in*oz of torque, rotating 360 degrees every 1.6 seconds (38 rpm - just a hair slower than a servo), at 5V, drawing 600mA at stall (free running at 52mA).

With a 7mm double-flat output shaft (avoid using the "D" output - it's not meant to take rotational load), and a built-in clutch (limiting at 60 in*oz, but easily "locked"), and built-in mounting screw holes, you can see why we like these motors!

Overall dimensions are 65mm (2.56") long x 22mm (0.867") x 18.5mm (0.73").
Mfr. # GM3

8) Summary

What were the results? Did you accomplish your goals?
If not, what challenges did you encounter? How may these be resolved?

The results of our project were very good. We did indeed get it to open during the time light was brighter outside than on the inside. This provided us with main goal of the alarm clock. During the night, the results were equally as good because it would shield users from the prying eyes of the outside world. I would say that we did accomplish our goals because our two main functions were fulfilled. Even though these two functions were fulfilled, we did run into some practical problems. We only made an automated system that would give us three modes of operation. The first operation would let light in all the time while the second mode of operation would shut off light completely. And the third opened the shutter 45 degrees. We did not really take into account that there may be times when we would need to open the shutters at night to see what is happening outside or to keep them closed during the day when we were to sleep in or when it was not necessary to wake up in the morning for that day. In order to deal with these issues, we could extend the project to have a manual override or a more programmed setting that would allow us to deal with these small errors. We could add a programmed clock and calendar that would allow us to do as we wished every day of the week or even month. We also felt that a normal manual override would be tedious and even defeat the purpose of having it automated and free of the hassle of going over to the shutter and turning it on or off. We could in the future extend the project by having the project be wireless using two arduino boards and the Xbee chip. With the wireless feature, we could manually override the shutter from the comfort of our beds or from anywhere in the room for that matter without walking all the way to the shutters. Overall, we feel that our project was a success because we did meet the needs that we set forth in the beginning of the quarter. Although the project is by no means perfect, we feel that we have made a huge step in creating a perfect shutter system that would be both practical and purposeful. Another problem that we ran into in the beginning phase of the shutter was to find a motor that would have enough power and torque to power such a shutter system. We found this motor in a toy car that we purchased from a hobby shop. This motor provided us with plenty of torque and speed restraints.