

LightCTL

Spencer Krum

August 8, 2010

Prerequisites

- ▶ Open Source Software Class
- ▶ Basic Circuits
- ▶ Basic C
- ▶ Basic Python

Overview of Topics

1. Dr. Wamser's Lab
2. The Circuit
3. The Two Programs
 - ▶ Dumb Teensy
 - ▶ Smart Teensy
4. Microcontrollers and You

Wamser Lab

- ▶ Dr. Wamser's group - SB1 324
- ▶ Organic Solar Cells - Artificial Photosynthesis
- ▶ Cells come out of the oven 2mm thick and 2.5cm square
- ▶ Cells are categorized by IV curves under constant illumination
- ▶ Lamp used was powered by batteries - poor consistency

Teensy++

- ▶ Teensy++ is an Arduino-like microcontroller
- ▶ Atmel AT90USB1286 - 8bit - 16Mhz - mini USB
- ▶ Cost - \$18 or \$24
- ▶ I have experience with it because of the UROV project
- ▶ Two types of output, PWM and digital I/O
- ▶ www.pjrc.com has Teensy and example code/tutorials
- ▶ Example code includes serial over USB
- ▶ Code written in C, gcc-avr package available as F/OSS

Power

- ▶ Lamp runs on 12V DC
- ▶ Formerly powered off batteries
- ▶ Now AC/DC transformer feeds into 12V synchronous regulator board
- ▶ Teensy outputs 8bit PWM signal to V_{adj} pin on regulator board
- ▶ Result is smooth control between 8 and 12.5 V DC

Input

- ▶ CdS (Cadmium Sulfide) photoresistor
- ▶ LDR is one half of a voltage divider spanning 5V
- ▶ Analog voltage is read at the middle of the divider
- ▶ Teensy has internal 10bit ADC
- ▶ External analog reference not used - option for future improvement

Circuit

- ▶ PWM is pulse width modulated
- ▶ PWM signal is hooked to V_{adj} pin on regulator board
- ▶ 5 digital I/O pins control 7 seg displays
- ▶ 1 I/O pin uses NAND IC(thanks Phil!) to multiplex across 2 displays
- ▶ 4 I/O pins use BCD to a BCD-7Segment display decoder to run 7-Seg
- ▶ Future version will use analog dial to display output

Two Versions

- ▶ In short: there are two versions of LightCTL, ideally they do the same thing
- ▶ One, Smart Teensy, runs a closed loop control system on the teensy
- ▶ The other, Dumb Teensy, runs a closed loop control system on the controlling computer in python

Teensy Programming Basics

- ▶ Teensy is a device with a little over 8K memory
- ▶ instead of int, use uint8_t
- ▶ Registers control everything
- ▶ Bit twiddling
- ▶ $\text{DDRB} |= (1 \ll 4) | (1 \ll 3);$

Control Loop

- ▶ Simple control loop uses PID
- ▶ P is for Proportion
- ▶ I is for Integral
- ▶ D is for Derivative

Serial

- ▶ PJRC supplied code has Teensy appear as serial device
- ▶ On Linux this looks like `/dev/ttyACM0`
- ▶ For the UROV greghaynes++ wrote a simple syntax for serial communication
- ▶ Python:
 - ▶ `import serial`
 - ▶ `x = serial.Serial('/dev/ttyACM0')`
 - ▶ `x.write('\x04\x03\xff\n')`
- ▶ First char indicates job, 4 is PWM, 7 is Sensor
- ▶ Second char indicates which pwm/sensor
- ▶ Third char (pwm only) indicates value
- ▶ Newline delimited

Whats next for LightCTL

- ▶ Combine Dumb Teensy and Smart Teensy
- ▶ Make it work with LabView
- ▶ Web interface/control
- ▶ Printed Circuit Board

You really can hack on a Microcontroller

- ▶ YOU! You really can.
 - ▶ There is a ton of example code for all kinds of microcontrollers
 - ▶ There are a ton of example circuits for everything imaginable
 - ▶ There are people in #cschat, irc.cat.pdx.edu, that can help you(I'm but one)
 - ▶ Often getting something simple to work can be empowering
 - ▶ Bart teaches a class
 - ▶ and finally...
- .

Circuit

- ▶ If the Physics Student Can ... you better be able to!

Thanks!

- ▶ Thanks for listening to me ramble about LightCTL
- ▶ You can get it at www.github.com/spencer-krum/LightCTL
- ▶ Thanks to Bart Massey for OSS class
- ▶ Thanks to Greg Haynes for writing a code base to work off
- ▶ Thanks to Erik Sanchez for the circuits stuff
- ▶ Thanks to Phil for the NAND/help with circuits
- ▶ Thanks to Devin Quirozoliver for lots of things, but mostly his vimrc and the beamer I hacked this one out of