CMSC 328: Embedded Operating Systems

Client-Server II: Pi Sockets

Our goal in this lab is to create a server program that runs on the Pi that allows clients to control the GPIO pins over the Internet. Your server should allow the client to set the direction of the GPIO pins, control output pins, query input pins, and configure pull up/down resistors.

Design

Design the communication protocol to be used between the client and server. What information needs to be sent to the server from the client? What does the server need to send back to the client? Should you support multiple clients? Should your client- server use TCP/IP or UDP/IP?

Implementation

Implement your server in C using sockets and pthreads. Your client can be implemented in Python, C, Java, or whatever language you'd like.

Evaluation

Evaluate the latency of using your client-server system – how long does it take to set an output high, or query an input? Compare the latency when both the client server are on the Pi and when the client is on another computer. Collect a large number of samples and report your findings.

Deliverables

lastname_lab8/

server.c	C server that uses TCP/IP or UDP/IP
client.{c,py}	A python or c client program that controls the Pi via TCP/IP
report.pdf	Describe the design of your messaging protocol; evaluation results.

Extra – PWM

In addition to driving output pins high or low, another way to control the pins is using something called Pulse Width Modulation[1] (PWM). Effectively, this technique turns a digital output channel into an analog one. This technique can vary the brightness of an LED between completely on and completely off. Servo motors[2] use a type of PWM to control their position. GPIO18 can perform hardware-based PWM, but not the other pins so we'll resort to software-based PWM. Your server should allow any of the GPIO outputs to be used with PWM by setting the pin, sleeping, clearing the pin, and sleeping again. The client should be able to control the duty cycle and period.

[1] https://learn.sparkfun.com/tutorials/pulse-width-modulation/pulse-width-modulation
[2] http://www.servocity.com/html/how_do_servos_work_.html