

Introducción a la Programación de Microcontroladores con Tecnologías Libres: Trabajo Final

Integrantes:

- Alan Matkorski
- Alejandro Sobko

Proyecto

Desarrollamos un circuito que en conjunto con una aplicación web, posibilita la manipulación del estado de una lámpara, el control de un televisor e información sobre la temperatura de la habitación donde el dispositivo se encuentra funcionando.

Hoy en día existe un gran número de dispositivos que cuentan con navegadores web (celulares, tablets, relojes, televisores, etc), por lo tanto al optar por una aplicación web como la interfaz para que el usuario interactúe con el dispositivo, se le está brindando la posibilidad de acceder a la misma desde un número de dispositivos que se encuentra en constante crecimiento.

Requerimientos

Para poder realizar este proyecto desde cero y ponerlo en marcha fueron necesarios los siguientes elementos:

- Arduino UNO
- Ethernet Shield
- Protoboard
- Cable USB A-B
- Cable Ethernet
- Cables para protoboard (20 macho a macho y 5 macho a hembra aprox.)
- Lámpara
- Alargue
- Televisor con control remoto
- Emisor Infrarrojo CQY99
- Receptor Infrarrojo TSOP4838
- Relé SRD-05VDC-SL-C

1. **Ethernet Shield:** es el componente responsable de servir la aplicación web mediante la cual es utilizada por el usuario a modo de interfaz para interactuar con el circuito.
2. **Receptor Infrarrojo:** es necesario a la hora de realizar la configuración del programa que se encargará interactuar con la tele. Este se utiliza para leer cuales son los códigos que el control remoto le envía a la tele.
3. **Emisor Infrarrojo:** este componente es el que una vez configurada la aplicación se encargará de enviarle las señales infrarrojas a la tele.
4. **Relé:** utilizado para controlar el estado en el que se encuentra la lámpara.
5. **Sensor de temperatura:** este componente tiene como output el voltaje proporcional a la temperatura en grados centígrados por lo tanto mediante la cuenta inversa es posible obtener la temperatura

Implementación

Una vez implementado el circuito definido en el esquemático, se puede proceder a la configuración y al posterior uso del dispositivo. Los dos sketches necesarios (Main.ino y Decoder.ino) para realizar dicha tarea se encuentran en el siguiente link: <http://www.mediafire.com/download/9oll07z6n01585t/Arduino.zip>

Con el circuito ya conectado, cargarle a la placa el sketch Decoder y abrir el Serial. Apuntar al receptor infrarrojo y presionar todos los números del control y el botón de encendido y apagado. Reemplazar cada uno los códigos hexadecimales ya configurados por el código correspondiente que se encuentra en el listado de constantes definidas en el otro sketch (Main.ino):

```
#define POWER      0xE0E040BF
#define ONE        0xE0E020DF
#define TWO        0xE0E0A05F
#define THREE      0xE0E0609F
#define FOUR       0xE0E010EF
#define FIVE       0xE0E0906F
#define SIX        0xE0E050AF
#define SEVEN      0xE0E030CF
#define EIGHT      0xE0E0B04F
#define NINE       0xE0E0708F
#define ZERO       0xE0E08877
```

El sketch Main tiene como dependencia la librería IRremote la cual actúa de driver para manejar los módulos infrarrojos. La misma se puede descargar desde <https://github.com/z3t0/Arduino-IRremote>. También será necesario modificar las líneas donde se envía la señal infrarroja y cambiar la función sendSAMSUNG por la función que corresponda de acuerdo a la marca de televisor con la que se esté trabajando. Las funciones disponibles son:

- sendRC5
- sendRC6
- sendNEC
- sendSony
- sendPanasonic
- sendJVC
- sendSAMSUNG
- sendWhynter
- sendAiwaRCT501
- sendLG
- sendSanyo
- sendMitsubishi
- sendDISH
- sendSharpRaw
- sendSharp
- sendDenon
- sendPronto

Si la marca de su televisor no se encuentra entre los métodos disponibles entonces la librería no da soporte para la misma.

Una vez configurado el sketch y con el circuito armado, cargarlo al Arduino.

Problemas

En el transcurso del desarrollo nos encontramos con problemas de distintos tipos, los más notorios fueron:

1. El webserver cae aleatoriamente después de un número aleatorio de requests realizadas al mismo.
2. Dificultad para utilizar la librería IRremote. Para desarrollar el prototipo utilizamos un televisor Samsung para realizar las pruebas. La librería IRremote no trae compatibilidad con el protocolo que utiliza esta marca en cuanto a sus controles remotos. Se pudo solucionar utilizando un fork de esta librería realizado por una persona la cual agregó el soporte para Samsung y algunas otras marcas además de brindar algunos bug fixes.

3. Limitado espacio. Arduino cuenta con una memoria flash de 32KB (ATmega328P), de los cuales 0.5KB es utilizado por el bootloader. Esto nos limitó un poco ya que nuestro sketch levanta un servidor web y el mismo entrega una aplicación web andando, por lo tanto el archivo se vuelve “pesado” para lo que es la capacidad de la memoria y esto nos limitó un poco a la hora de escribir código.