

Arduinoponia

Sistema de cuidado y control de plantas en interior.

Julio 14, 2016

Integrantes: Nazareno Castro y Damián Mantuano.

UNQ – TPI. Seminario: Introducción a la Programación de Microcontroladores con Tecnologías Libres

Descripción:

Arduinoponia: Su nombre es una combinación del nombre de la placa Arduino y “ponos” del griego, que significa trabajo o labor.

La misión de Arduinoponia es de ofrecer una serie de sensores (Humedad y temperatura ambiente, humedad del sustrato y luminosidad) que además de ofrecer información para realizar seguimientos sobre el desarrollo de la planta, dispara eventos, pudiendo así configurar horario de encendido y apagado del sistema de iluminación, umbral de humedad del sustrato, por debajo del cual se activará el sistema de riego.

Logrando de esta manera, que el desarrollo de la planta sea óptimo y requiriendo muy pocos cuidados. Siendo ideal para cuando no hay quien se ocupe de la planta por un fin de semana largo o época de vacaciones.

El sistema de Arduinoponia consta de una caja de madera rectangular de aproximadamente 25cm de ancho, 25cm de profundidad y 70cm de alto. Abriendo la puerta, podemos encontrar en su interior la base con maceta donde se coloca la planta, la cual cuenta con una manguera perforada en todo su diámetro interno, la cual está conectada con el sistema de riego, brindando así un riego por goteo de forma pareja. Manteniendo de esta forma la humedad del sustrato y conservando los nutrientes del mismo. Un poco más arriba se encuentra la lámpara de bajo consumo que ilumina la planta para brindarle la cantidad y calidad de luz necesaria para el desarrollo de la planta. El interior del cajón se encuentra recubierto en papel metalizado para reflejar la luz en todas direcciones.

El cerebro de Arduinoponia cuenta con una memoria SD en la cual se encuentra un archivo de configuración (settings.ini) donde se debe configurar el horario de encendido y apagado de la luz en formato 24hs. También se debe indicar el nivel nivel de humedad del sustrato por debajo del cual se activará el sistema de riego. También se encuentra otro archivo (data.log) el cual funciona como registro histórico de la planta, el cual cuenta con la información de los sensores y sirve para llevar un control del progreso de la planta en el tiempo. Así como también registrar los eventos (Encendido/Apagado de luces, Activado del sistema de riego, etc)

Fecha: 14/07/2016

Hora: 18:39:53

Humedad Ambiente: 71.10 %

Temperatura Ambiente: 22.20° C

Humedad sustrato:60%

Sensor de Luz: Luz Apagada

Estado de la luz: Apagada

Ejemplo de data.log

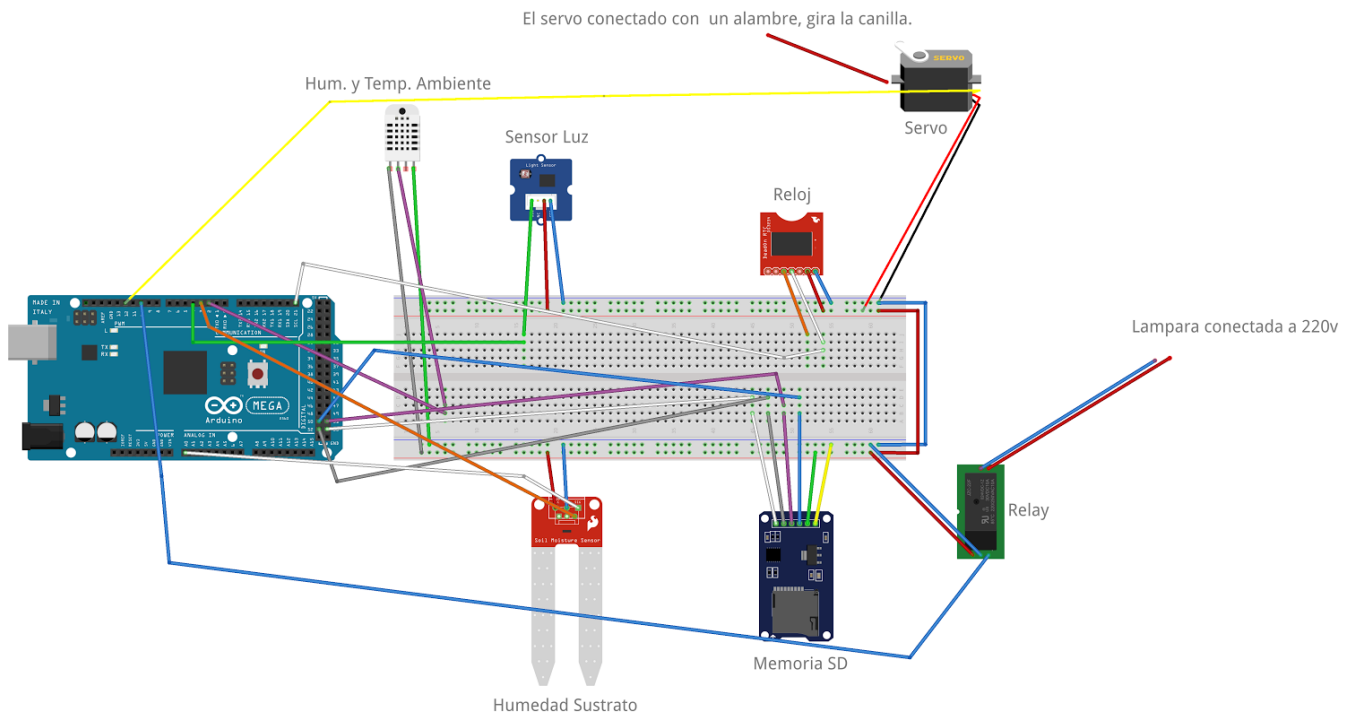
[luz_horario_encendido=22]

[luz_horario_apagado=10]

[riego_umbral=50]

Ejemplo de data.log

Esquema:



Elementos utilizados:

Componentes electrónicos:

- Modulo Sensor De Luz Con Ldr Fotoresistor \$69.90 [[LINK](#)]
- Protoboard De 830 Puntos \$129.90 [[LINK](#)]
- Sensor Humedad Relativa Y Temperatura Dht22 \$149.90 [[LINK](#)]
- Modulo Sensor De Humedad De Suelo Tierra \$79.90 [[LINK](#)]
- Arduino Mega2560 \$299,90 [[LINK](#)]
- Módulo Micro Sd Card 5v Con Adaptador 3v3 \$69.90 [[LINK](#)]
- Reloj Tiempo Real Rtc Ds3231 Eeprom 24c32 \$99.90 [[LINK](#)]
- Kit 40 Cables Para Protoboard Macho Hembra \$79.90 [[LINK](#)]
- Kit 65 Cables Para Protoboard Macho Macho \$79.90 [[LINK](#)]
- Mini Servo Tower Pro Sg90 1.5kg \$80 [[LINK](#)]
- Cables interiores de un cable utp.

Otros componentes:

- Madera Fibrofacil \$400 [EASY]
- Bisagras \$20 [FERRETERIA/[ML](#)]
- Tornillos \$10 FERRETERIA/[ML](#)]
- Fondo de Madera Blanco [FERRETERIA/ML]
- Canilla de ½ \$40 [FERRETERIA/[ML](#)]
- Portalamparas de 220V [FERRETERIA/ML]
- Lampara Bajo Consumo de xW \$400 [FERRETERIA/[ML](#)]
- Manguera de ½ \$15 [FERRETERIA/[ML](#)]
- 2 Clip [FERRETERIA/ML]
- Manguera acuario x 1.5mt \$20 [ACUARIO/[ML](#)]
- Conector T para manguera acuario \$5 [ACUARIO/[ML](#)]

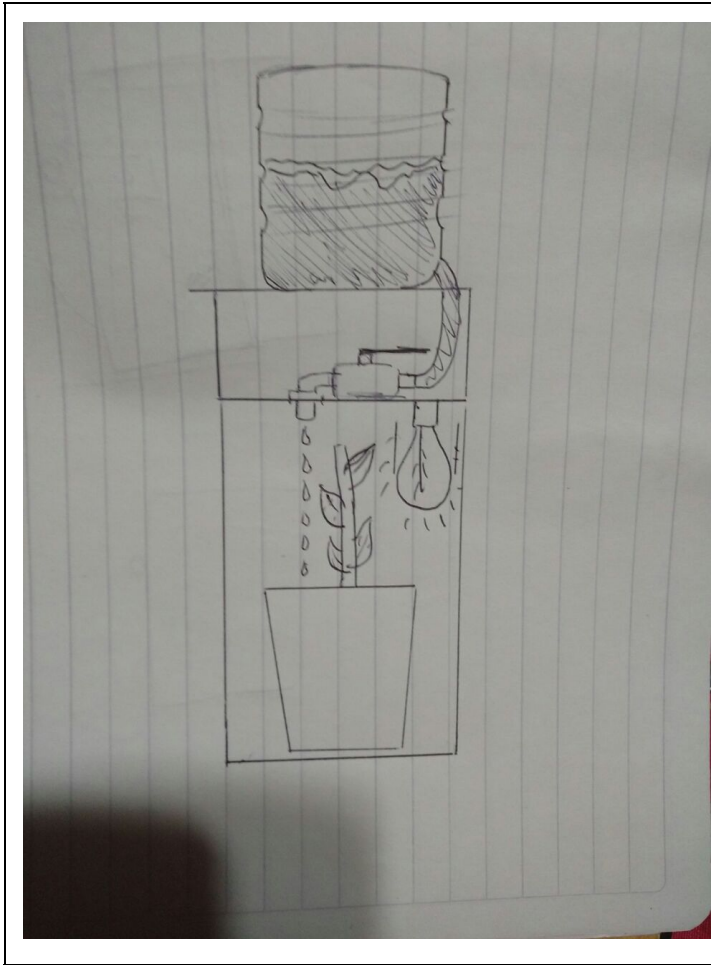
Problemas:

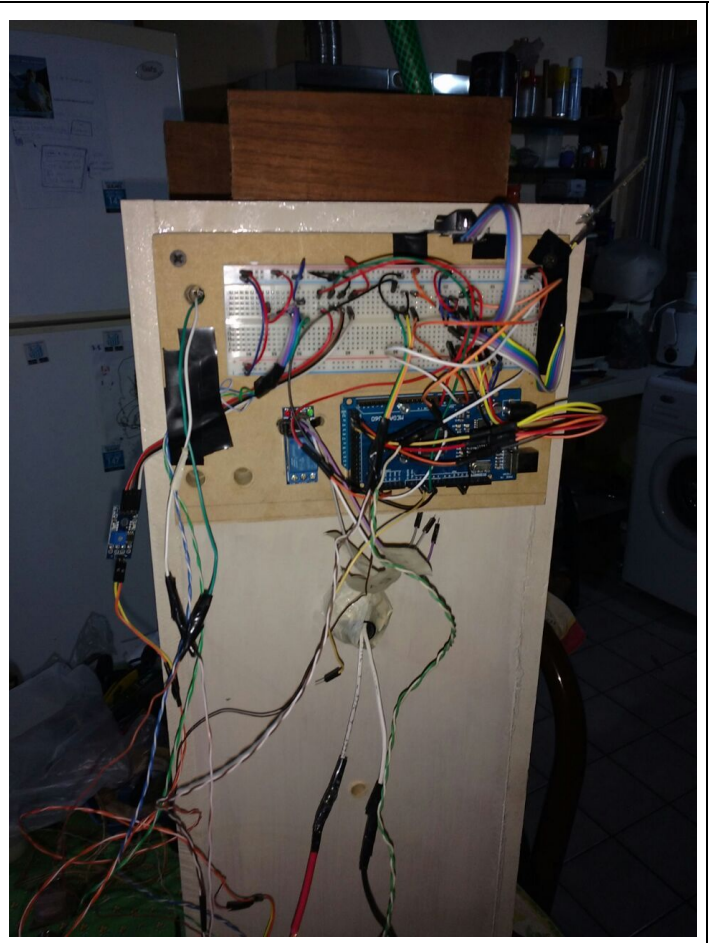
Nos encontramos con algunos problemas al intentar encarar este proyecto. Uno de ellos fue el sistema de riego. El cual inicialmente habíamos pensado en hacerlo utilizando una bomba de agua eléctrica, la cual nos resultó algo cara y a su vez también nos parecía más interesante la perspectiva de intentar algo más ingenioso. Así fue como terminamos usando un servo motor para abrir y cerrar la llave de la canilla.

Inicialmente empezamos trabajando con [Firmata](#) en el Arduino y [Johnny-five](#) corriendo en un servidor, permitiendo así poder controlar los parámetros de riego e iluminación sin tener que reprogramar el Arduino. Pero nos encontramos con el problema de que para algunos de los módulos que estábamos utilizando no había librerías listas para utilizar, como por ejemplo para el DHT22. Fue por esto y porque la necesidad de tener el Arduino conectado a una computadora constantemente nos resultaba muy limitante. Por eso decidimos dejar de lado Johnny-five para este proyecto y colocar todo el código en Arduino. Utilizando la memoria SD para guardar el log y leer la configuración, manteniendo el dinamismo.

Otro problema que encontramos fue la iluminación. Inicialmente la idea era utilizar LEDs, pero dado que las plantas requieren de una gran cantidad de luz, necesitábamos utilizar una gran cantidad de leds regulares (aprox 300) o utilizar LEDs de alta potencia (3w), los cuales son caros y requieren de un sistema de ventilación ya que generan mucho calor. Estas razones fueron suficientes para que optáramos por utilizar una lámpara de 100w manejada con un relé para resolver el problema de la iluminación de una forma efectiva.

Fotos del proceso:







Repositorio:

<https://github.com/smarbos/arduinoponia>

Todo:

- Utilizar modulo WIFI, implementar protocolo MQTT.
- Enviar alertas cuando queda poca agua en tanque.
- Enviar reporte con estado de la planta
- Implementar sistema de ventilación