

Proyecto final: Jardín Automatizado usando arduino

Adrián Avilés Flores, B80835 y Steven Mora Barboza, B95109

I. INTRODUCCIÓN

En el siguiente informe se presenta el desarrollo del proyecto de jardín automatizado con Arduino. Este proyecto consistió en crear un ambiente controlado para cuidar plantas, utilizando un sensor de temperatura y otro de humedad especial para medir la humedad de la tierra. Además, se conectó un abanico y una válvula de agua al Arduino para que se enciendan automáticamente cuando sea necesario enfriar el ambiente o regar las plantas. Todo esto se puede controlar y monitorear a través de una pantalla conectada al Arduino y de la plataforma de ThingsBoard y es importante mencionar que este proyecto ha sido inspirado a partir del proyecto "Plantduino Green house"[1]. Durante el desarrollo del proyecto, se llevaron a cabo pruebas exhaustivas para garantizar la precisión de las mediciones y el funcionamiento adecuado de los componentes electrónicos. Además, se realizaron ajustes y mejoras en el proyecto para garantizar la protección del Arduino y los componentes electrónicos contra la humedad, para así dar como resultado un exitoso comportamiento del sistema según se esperaba.

II. OBJETIVO GENERAL

Explorar las posibilidades que brinda la tecnología para el cuidado de plantas y mejorar la eficiencia y calidad del cuidado de las mismas. Para ello, se busca crear un ambiente controlado mediante la utilización de componentes electrónicos y software programable, que permitan medir y controlar variables importantes como la temperatura y la humedad, y actuar sobre el ambiente cuando sea necesario mediante el encendido de un abanico para enfriar el ambiente o la activación de una válvula para regar las plantas.

III. OBJETIVOS ESPECIFICOS

- Diseñar y construir un ambiente controlado adecuado para el crecimiento de plantas en macetas.
- Seleccionar y conectar los componentes electrónicos necesarios para medir y controlar la temperatura y la humedad del ambiente y del sustrato de las plantas.
- Programar el Arduino para recibir y procesar los datos de los sensores, y actuar sobre el ambiente mediante el encendido de un abanico o la activación de una válvula de agua.
- Implementar una pantalla que muestre la información sobre las variables medidas y el último riego realizado.
- Integrar el Arduino con la plataforma de monitoreo de datos ThingsBoard para el registro y análisis en tiempo real de los datos de temperatura, humedad y riego.
- Realizar pruebas para verificar el correcto funcionamiento del sistema y hacer ajustes y mejoras según sea necesario.

IV. ALCANCES

- Diseño y construcción de un ambiente controlado: El proyecto permitirá diseñar y construir un ambiente controlado que permita mantener las condiciones adecuadas de temperatura y humedad para el crecimiento de plantas en macetas.
- Medición y control de variables importantes: El sistema permitirá medir y controlar variables importantes como la temperatura y la humedad del ambiente y del sustrato de las plantas, lo que permitirá ajustar el ambiente según las necesidades de las plantas.
- Automatización del riego y enfriamiento: El proyecto permitirá automatizar el riego y el enfriamiento del ambiente mediante la activación de una válvula de agua y un abanico, respectivamente, lo que permitirá mantener un ambiente óptimo para el crecimiento de las plantas.
- Visualización de información: El proyecto permitirá visualizar información importante sobre las variables medidas y el último riego realizado en una pantalla, lo que permitirá monitorear el estado de las plantas y el funcionamiento del sistema.
- Monitoreo de datos en tiempo real: El proyecto permitirá el monitoreo de datos de temperatura, humedad y riego en tiempo real mediante la plataforma de monitoreo de datos ThingsBoard, lo que permitirá analizar el comportamiento del sistema y realizar mejoras según sea necesario.
- Aprendizaje y experiencia en el uso de tecnología: El proyecto permitirá aprender y experimentar con la utilización de tecnología para el cuidado de plantas, lo que permitirá aplicar estos conocimientos en otros proyectos similares en el futuro.
- Y como último, el sistema diseñado es escalable y permite su expansión a futuro. Esto significa que el proyecto puede ser ampliado o mejorado en el futuro, agregando nuevos sensores para medir otras variables ambientales, así como nuevos actuadores para realizar otras acciones en el ambiente controlado

V. JUSTIFICACION

Ante la necesidad de crear ambientes controlados para el crecimiento de plantas en zonas urbanas y en espacios reducidos. La urbanización ha generado la disminución de los espacios verdes y la dificultad de obtener alimentos frescos y saludables, por lo que se ha popularizado la idea de cultivar plantas en espacios reducidos, como balcones, terrazas, patios y otros lugares. Sin embargo, mantener un ambiente adecuado para el crecimiento de las plantas puede ser un desafío, ya que se requiere controlar la temperatura, la humedad y otros factores ambientales para obtener un crecimiento saludable. Además, la automatización del riego y el enfriamiento puede permitir

un ahorro de tiempo y energía en el cuidado de las plantas. En este contexto, la utilización de tecnología como Arduino y sensores especializados permite el diseño y construcción de sistemas automatizados y controlados para el cuidado de plantas en ambientes reducidos, lo que permite obtener plantas saludables y frescas en un espacio reducido. Además, el uso de tecnología permite la obtención de datos en tiempo real y el monitoreo remoto de las variables ambientales, lo que permite un mejor seguimiento del estado de las plantas y el funcionamiento del sistema.

VI. MARCO TEÓRICO

VI-A. Arduino Uno (ATmega 328p)

El Arduino UNO es una placa de desarrollo basada en el microcontrolador ATmega328P. Este microcontrolador es un microprocesador AVR de 8 bits desarrollado por Atmel. Este es un dispositivo con arquitectura RISC (Reduced Instruction Set Computing), lo que significa que tiene un conjunto reducido de instrucciones, pero altamente optimizadas para realizar tareas específicas. Esto le permite ser muy eficiente en términos de velocidad y consumo de energía. El ATmega328P cuenta con 32 KB de memoria flash para almacenar el programa, 2 KB de SRAM y 1 KB de EEPROM para almacenar datos. También cuenta con 14 pines de entrada/salida digitales, 6 pines de entrada analógica, una conexión USB y una conexión de alimentación de 5V. [2].

5.1. Pin-out

Figure 5-1. 28-pin PDIP

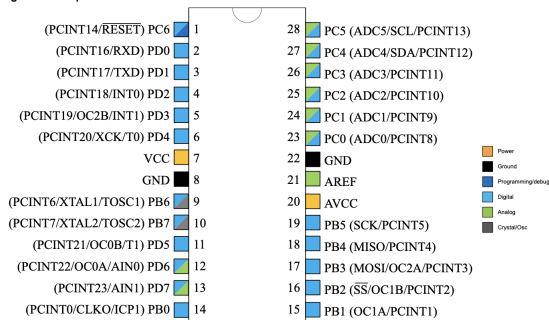


Figura 1. Diagrama de pines Arduino Uno [3].

En el caso del laboratorio, se utilizarán los pines analógicos para conectar el sensor de temperatura y el de humedad, los cuales corresponden al A0 y el A1.

A continuación el diagrama de bloques del MCU:

Block Diagram

Figure 4-1. Block Diagram

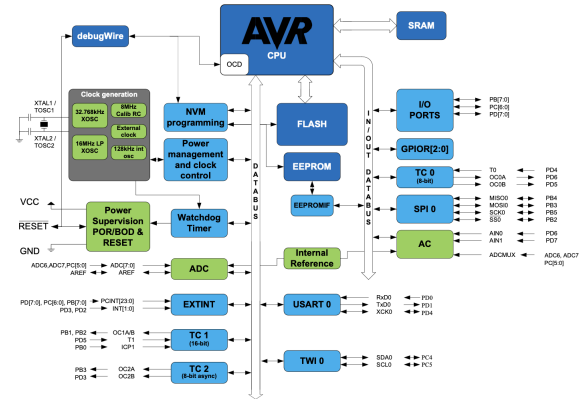


Figura 2. Diagrama de bloques del Arduino Uno [3].

Entre los componentes a usar según el diagrama de bloques, se utilizará el conector USB para programar y alimentar el Arduino, regulador de tensión, que convierte la alimentación externa de hasta 12V en una alimentación de 5V estable para el microcontrolador, puertos de entrada/salida (I/O), se utilizan para conectar sensores, actuadores y otros componentes externos al Arduino. ADC conversor analógico-digital que convierte la señal analógica en una señal digital para su procesamiento por el microcontrolador. USART: bloque de comunicación serial que permite la transmisión de datos entre el Arduino y una PC. Crystal Oscillator: proporciona una frecuencia de reloj para el microcontrolador.

32.1. Absolute Maximum Ratings

Table 32-1. Absolute Maximum Ratings

| | |
|--|--------------------------------|
| Operating Temperature | -55°C to +125°C |
| Storage Temperature | -65°C to +150°C |
| Voltage on any Pin except RESET with respect to Ground | -0.5V to V _{CC} +0.5V |
| Voltage on RESET with respect to Ground | -0.5V to +13.0V |
| Maximum Operating Voltage | 6.0V |
| DC Current per I/O Pin | 40.0mA |
| DC Current V _{CC} and GND Pins | 200.0mA |

Figura 3. Especificaciones eléctricas del Arduino UNO [3].

Además, se usará como fuente de energía para el Arduino su entrada por cable USB de 5 V, esto alimentará tanto a los Relays, como los sensores de temperatura y humedad. En este caso, la pantalla será conectada en los pines A4 y A5 para los puertos SDA y SCL respectivamente y los relays al pin 2 y 4.

VI-B. Periféricos utilizados

VI-B1. Virtual Serial Port Tools: Virtual Serial Port Tools es un software que permite crear puertos serie virtuales en un ordenador, los cuales pueden ser usados por programas como si fueran puertos serie reales, en este caso, para hacer la comunicación entre COM4 (Arduino) y COM3 (PC).

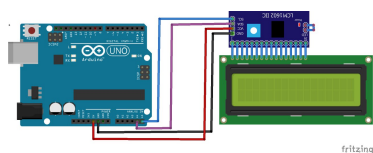


Figura 4. Esquemático de la pantalla LCD i2c

En este caso, se conectan los puertos SDA y SCL a los pines A4 y A5 según se menciona en la guía del componente.

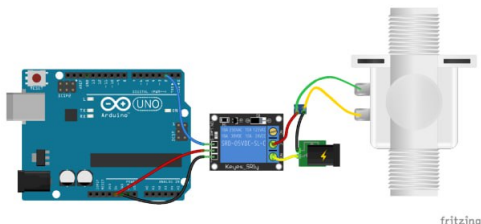


Figura 5. Esquemático de la válvula solenoide

La válvula solenoide, como mencionado anteriormente, se conecta a una fuente de 12 V y un relay que será el que indica cuando activar el abanico,

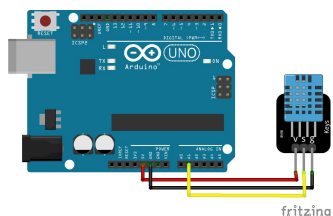


Figura 6. Esquemático del sensor DHT11

El sensor DHT11 está conectado al puerto analógico A1 y a la fuente del Arduino de 5 V.

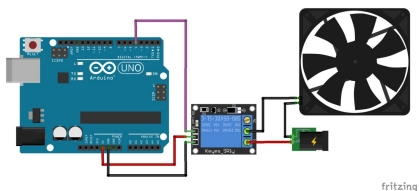


Figura 7. Esquemático de la conexión del abanico

El abanico está conectado a un relay, y este al puerto digital 4, de esta forma el Arduino controla cuando el abanico se enciende o no, por medio del relay, de la misma forma el abanico está conectado a una fuente de 12 V externa debido a que no puede proporcionar 12 V.

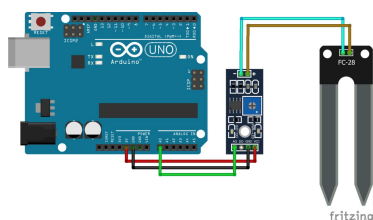


Figura 8. Esquemático del sensor de humedad

Finalmente, el sensor de humedad está conectado al puerto analógico A0, y a la fuente de 5 V del Arduino

A continuación la lista de componentes utilizados con sus respectivos precios en dólares:

| Nombre | Cantidad | Precio en \$ |
|-------------------------------|----------|--------------|
| Arduino UNO | 1 | 35 |
| DHT11 | 1 | 3 |
| Sensor Humedad | 1 | 2 |
| Cables | - | 5 |
| Set de 2 Relays | 1 | 5 |
| Abanico | 1 | 2 |
| Tierra | 3kg | 7 |
| Madera | 2 U | 7 |
| Tornillos | 20 | 8 |
| Tubos PVC | 4 | 6 |
| Manguera | 1 | 4 |
| Accesorios para tubos y tubos | 8 | 7 |
| Válvula Solenoide | 1 | 9 |
| Protoboard | 1 | 5 |
| Pantalla LCD i2C | 1 | 6 |
| Cinta pegajosa | 1 | 2 |
| Teflón | 1 | 1 |

TABLA I
PRECIO DE LOS COMPONENTES

VI-C. Análisis de Resultados

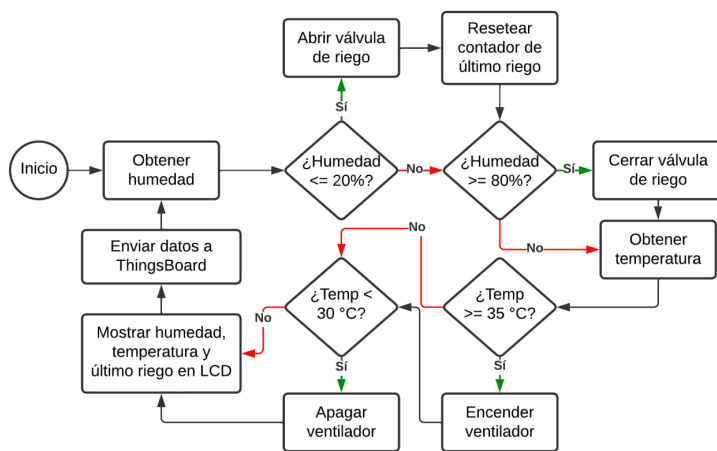


Figura 9. Diagrama de flujo del funcionamiento del sistema

En la Figura 9, se puede apreciar el comportamiento del sistema, en donde al iniciarse primeramente se lee la señal analógica del sensor de humedad, si la humedad tiene un valor menor o igual de 20 %, la válvula se abrirá para regar las plantas, y a la vez, el contador que almacena la última vez de regadas las plantas, se vuelve cero, de esta manera, se llega al siguiente cuadro decisivo, donde si la humedad supera un 80 %, la válvula será cerrada y de inmediato se lee la temperatura, si esta es mayor o igual que 35 grados Celsius, se encenderá el abanico, y de manera contraria, si la temperatura es menor a 30 grados Celsius, el abanico se apagará. Cabe destacar que los datos de la humedad, temperatura y último riego siempre se están imprimiendo en la pantalla del Arduino, y además, imprimiéndose en la plataforma Thingsboard.

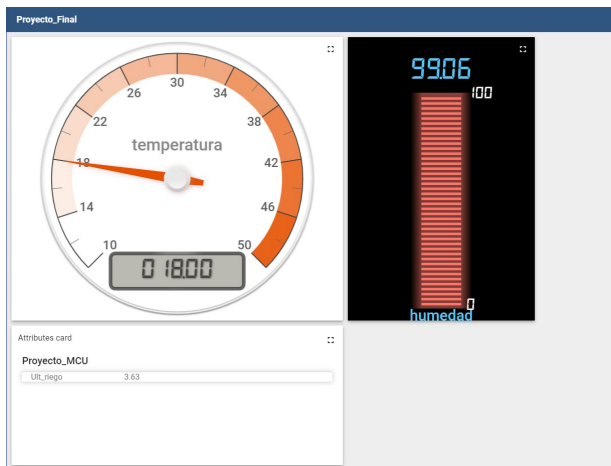


Figura 10. Funcionamiento del Thingsboard

En la Figura 10, se puede observar como los datos de la humedad, temperatura y última vez regada, pueden ser mostrados en tiempo real en la plataforma Thingsboard de forma gráfica y limpia como lo esperado.

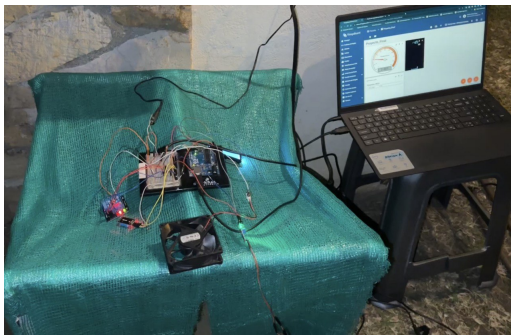


Figura 11. Prueba del funcionamiento del sistema

En la Figura 11, se puede apreciar el funcionamiento final del sistema, donde la pantalla, los sensores y el Thingsboard están encendidos y corriendo correctamente, y el invernadero cerrado, de manera que se pueda crear el ambiente controlado dentro del mismo.

El video completo donde se demuestra el funcionamiento de este sistema se encuentra en el siguiente link: https://www.youtube.com/watch?v=Oen8aWoe298&ab_channel=StevenMora

VI-D. Conclusiones

- La automatización del cuidado de las plantas mediante el uso de tecnología puede permitir un ahorro significativo de tiempo y esfuerzo, lo que lo hace una solución conveniente para personas que tienen poco tiempo para dedicar al cuidado de las plantas.
- El uso de Arduino y sensores especializados permite el diseño y construcción de sistemas personalizados y escalables para el cuidado de plantas en ambientes reducidos.
- La monitorización en tiempo real de las variables ambientales como la humedad y la temperatura puede

permitir una respuesta rápida ante cambios en el ambiente controlado, lo que contribuye a mantener un ambiente óptimo para el crecimiento de las plantas.

- El uso de tecnología para el cuidado de plantas en espacios reducidos puede contribuir a la obtención de alimentos frescos y saludables en zonas urbanas, lo que promueve la alimentación sana y el bienestar en la población.
- El orden de los cables es sumamente importante para evitar circuitos abiertos o corto circuitos y así no llegar, en el peor de los casos, a quemar algún dispositivo.

VI-E. Observaciones

- Es importante tener en cuenta el mantenimiento periódico del sistema para garantizar su correcto funcionamiento y prolongar su vida útil.
- En caso de utilizar materiales de bajo costo, es importante asegurarse de que sean resistentes y duraderos para evitar posibles fallos del sistema.
- Se debe tener precaución en la ubicación de los sensores para evitar posibles interferencias que puedan afectar la lectura de las variables ambientales.

VI-F. Recomendaciones

- Se puede considerar la incorporación de una batería de respaldo o un sistema de alimentación ininterrumpida (UPS) para asegurar el funcionamiento del sistema en caso de cortes de energía eléctrica.
- Se puede explorar la posibilidad de integrar otros sensores para medir otras variables importantes para el crecimiento de las plantas, como la intensidad de luz y la concentración de CO₂.
- Se puede considerar la integración de un sistema de notificación o alerta para informar al usuario sobre posibles problemas o situaciones críticas que requieran atención inmediata.
- Se debe tener en cuenta que si algún periférico no está funcionando de manera adecuada, o el mismo Arduino, puede ser debido a que exista algún corto circuito debido a la gran cantidad de cables que existe, esto se menciona porque ese fue un desafío importante al momento de desarrollar este laboratorio.

El repositorio con el desarrollo de este proyecto se encuentra en: <https://github.com/Stevenlmb/Lab-Microcontroladores/tree/Proyecto>

VI-G. Bibliografía

REFERENCIAS

- [1] Clover. (2011) Plantduino greenhouse. [Online]. Available: <https://www.instructables.com/Backyard-Automated-Greenhouse/?lang=es>
- [2] A. A. Steven Mora, "Laboratorio #3: Arduino: Gpio,adc y comunicaciones," 2023.
- [3] Atmel. (2016) 8-bit avr microcontrollers atmega328/p. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/241077/ATMEL/ATMEGA328P.html>



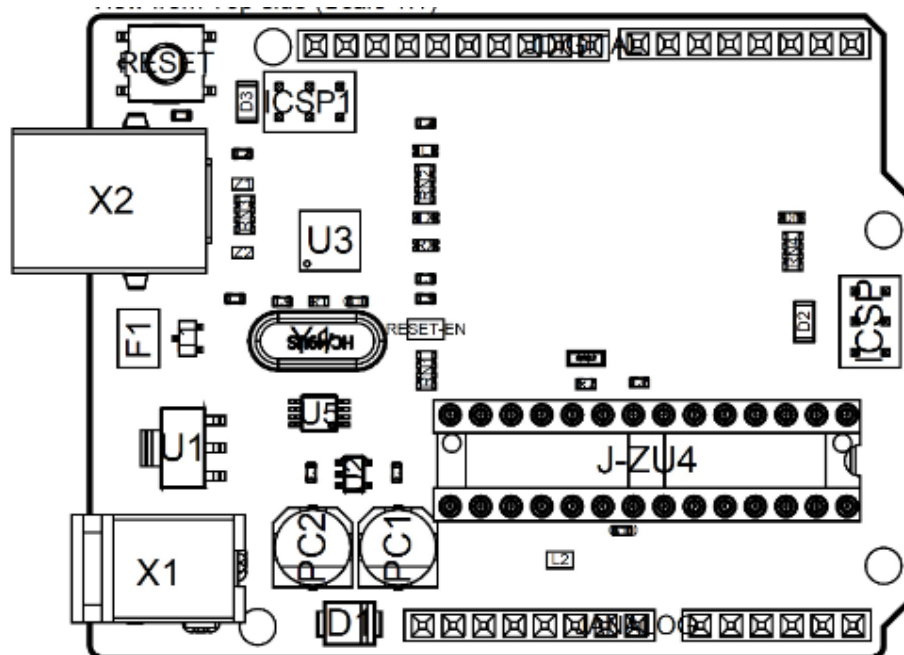
2.2 Power Consumption

| Symbol | Description | Min | Typ | Max | Unit |
|---------|--|-----|-----|-----|------|
| VINMax | Maximum input voltage from VIN pad | 6 | - | 20 | V |
| VUSBMax | Maximum input voltage from USB connector | | - | 5.5 | V |
| PMax | Maximum Power Consumption | - | - | xx | mA |

3 Functional Overview

3.1 Board Topology

Top view



Board topology

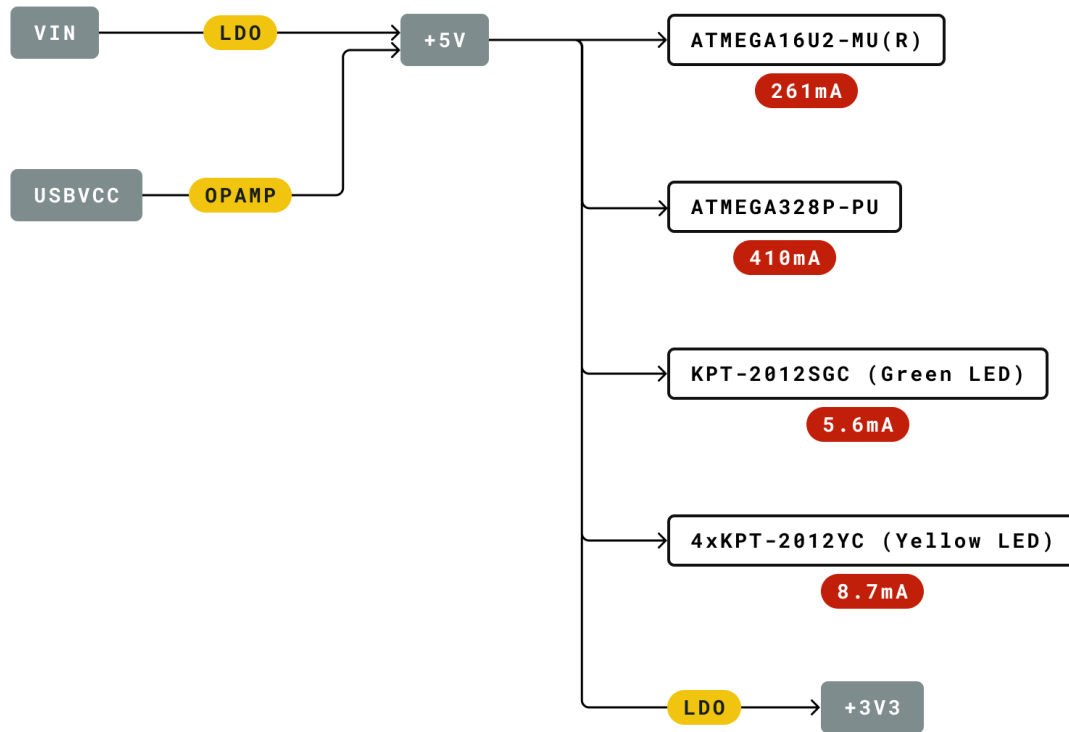
| Ref. | Description | Ref. | Description |
|-------|--------------------------------|-------|---------------------------------------|
| X1 | Power jack 2.1x5.5mm | U1 | SPX1117M3-L-5 Regulator |
| X2 | USB B Connector | U3 | ATMEGA16U2 Module |
| PC1 | EEE-1EA470WP 25V SMD Capacitor | U5 | LMV358LIST-A.9 IC |
| PC2 | EEE-1EA470WP 25V SMD Capacitor | F1 | Chip Capacitor, High Density |
| D1 | CGRA4007-G Rectifier | ICSP | Pin header connector (through hole 6) |
| J-ZU4 | ATMEGA328P Module | ICSP1 | Pin header connector (through hole 6) |
| Y1 | ECS-160-20-4X-DU Oscillator | | |



3.2 Processor

The Main Processor is a ATmega328P running at up to 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.

3.3 Power Tree



Legend:

- | | | |
|-----------|-------------|-----------------|
| Component | Power I/O | Conversion Type |
| | Max Current | Voltage Range |

Power tree



4 Board Operation

4.1 Getting Started - IDE

If you want to program your Arduino UNO while offline you need to install the Arduino Desktop IDE [1] To connect the Arduino UNO to your computer, you'll need a Micro-B USB cable. This also provides power to the board, as indicated by the LED.

4.2 Getting Started - Arduino Web Editor

All Arduino boards, including this one, work out-of-the-box on the Arduino Web Editor [2], by just installing a simple plugin.

The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow [3] to start coding on the browser and upload your sketches onto your board.

4.3 Getting Started - Arduino IoT Cloud

All Arduino IoT enabled products are supported on Arduino IoT Cloud which allows you to Log, graph and analyze sensor data, trigger events, and automate your home or business.

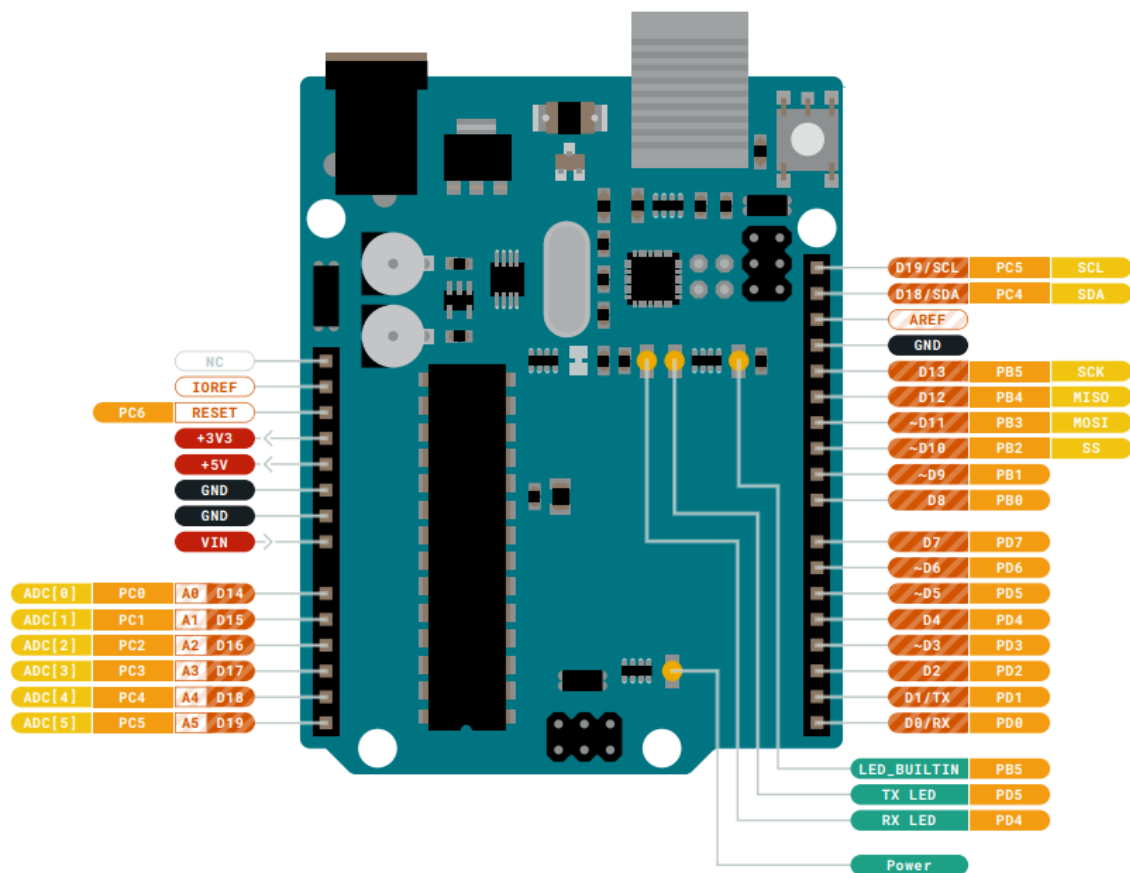
4.4 Sample Sketches

Sample sketches for the Arduino XXX can be found either in the "Examples" menu in the Arduino IDE or in the "Documentation" section of the Arduino Pro website [4]

4.5 Online Resources

Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on ProjectHub [5], the Arduino Library Reference [6] and the online store [7] where you will be able to complement your board with sensors, actuators and more

5 Connector Pinouts



Pinout

Introduction

The Atmel® picoPower® ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

Feature

High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller Family

- Advanced RISC Architecture
 - 131 Powerful Instructions
 - Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 32KBytes of In-System Self-Programmable Flash program Memory
 - 1KBytes EEPROM
 - 2KBytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data Retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Atmel® QTouch® Library Support
 - Capacitive Touch Buttons, Sliders and Wheels
 - QTouch and QMatrix® Acquisition
 - Up to 64 sense channels

- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Two Master/Slave SPI Serial Interface
 - One Programmable Serial USART
 - One Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - One On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V
- Temperature Range:
 - -40°C to 105°C
- Speed Grade:
 - 0 - 4MHz @ 1.8 - 5.5V
 - 0 - 10MHz @ 2.7 - 5.5V
 - 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
 - Active Mode: 0.2mA
 - Power-down Mode: 0.1µA
 - Power-save Mode: 0.75µA (Including 32kHz RTC)