

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232<br/>B) Conversión analógica / digital - P.W.M.<br/>Aplicaciones con Arduino</b> |

## **TRANSMISION EN PARALELO Y EN SERIE**

La comunicación en los medios informáticos se realiza de dos maneras

### **PARALELO**

La transmisión de datos entre ordenadores y terminales mediante cambios de corriente o tensión por medio de cables o canales; la transferencia de datos es en paralelo si transmitimos un grupo de bits sobre varias líneas o cables

En la transmisión de datos en paralelo cada bit de un carácter se transmite sobre su propio cable. En la transmisión de datos en paralelo hay un cable adicional en el cual enviamos una señal llamada strobe ó reloj; esta señal le indica al receptor cuando están presentes todos los bits para que se puedan tomar muestras de los bits o datos que se transmiten y además sirve para la temporización que es decisiva para la correcta transmisión y recepción de los datos.

La transmisión de datos en paralelo se utiliza en sistemas digitales que se encuentran colocados unos cerca del otro, además es mucho más rápida que la serie, pero además es mucho más costosa.

Todos los bits se transmiten simultáneamente, existiendo luego un tiempo antes de la transmisión del siguiente bloque.

Este tipo de transmisión tiene lugar en el interior de una maquina o entre maquinas cuando la distancia es muy corta. La principal ventaja de esto modo de transmitir datos es la velocidad de transmisión y la mayor desventaja es el costo.

También puede llegar a considerarse una transmisión en paralelo, aunque se realice sobre una sola línea, al caso de multiplexación de datos, donde los diferentes datos se encuentran intercalados durante la transmisión.

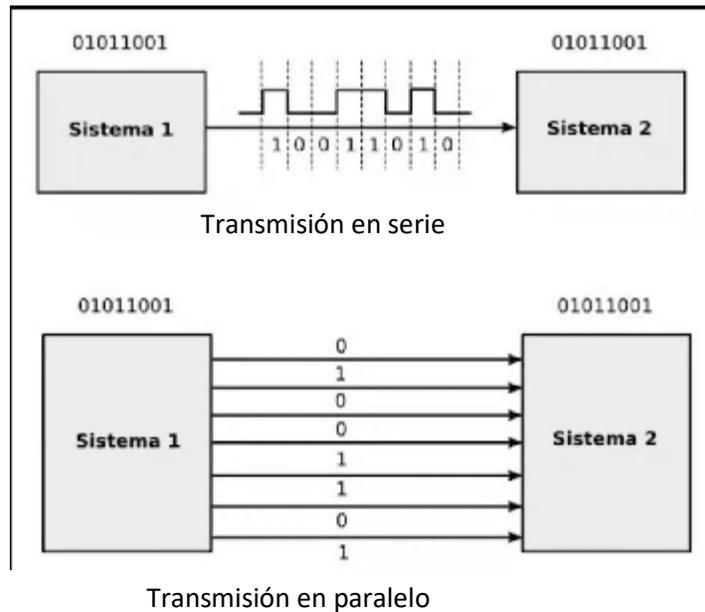
### **SERIE**

En este caso los n bits que componen un mensaje se transmiten uno detrás de otro por la misma línea.

Este tipo de transmisión se utiliza a medida que la distancia entre los equipos aumenta a pesar que es más lenta que la transmisión paralelo y además menos costosa. Los transmisores y receptores de datos serie son más complejos debido a la dificultad en transmitir y recibir señales a través de cables largos.

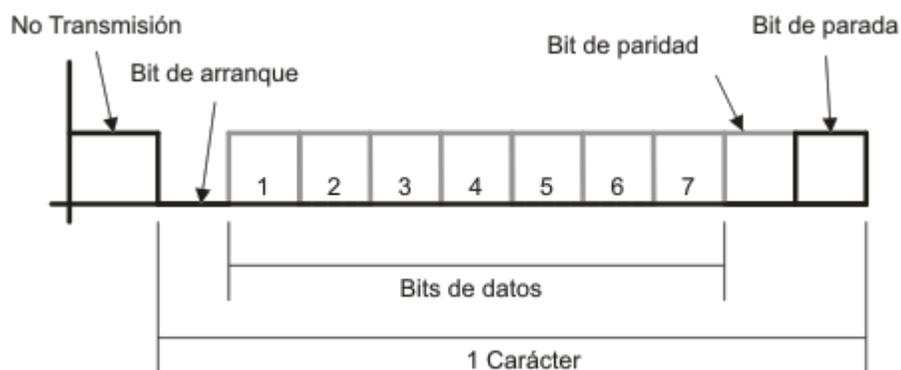
La conversión de paralelo a serie y viceversa la llevamos a cabo con ayuda de registros de desplazamiento.

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232</b><br><b>B) Conversión analógica / digital - P.W.M.</b><br><b>Aplicaciones con Arduino</b> |



### Transmisión asincrónica

Es también conocida como Start /Stop. Requiere de una señal que identifique el inicio del carácter y a la misma se la denomina bit de arranque. También se requiere de otra señal denominada señal de parada que indica la finalización del carácter o bloque.



Generalmente cuando no hay transmisión, una línea se encuentra en un nivel alto. Tanto el transmisor como el receptor, saben cual es la cantidad de bits que componen el carácter (en el ejemplo son 7). Los bits de parada son una manera de fijar qué delimita la cantidad de bits del carácter y cuando se transmite un conjunto de caracteres, luego de los bits de parada existe un bit de arranque entre los distintos caracteres.

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232</b><br><b>B) Conversión analógica / digital - P.W.M.</b><br><b>Aplicaciones con Arduino</b> |

A pesar de ser una forma comúnmente utilizada, la desventaja de la transmisión asincrónica es su bajo rendimiento, puesto que como en el caso del ejemplo, el carácter tiene 7 bits pero para efectuar la transmisión se requieren 10. O sea que del total de bits transmitidos solo el 70% pertenecen a datos.

### Norma Serie RS232

Se trata de una comunicación serie bidireccional (line driver and receiver)

Cuando se dice nivel TTL, se refiere a toda la electrónica digital con +5V.

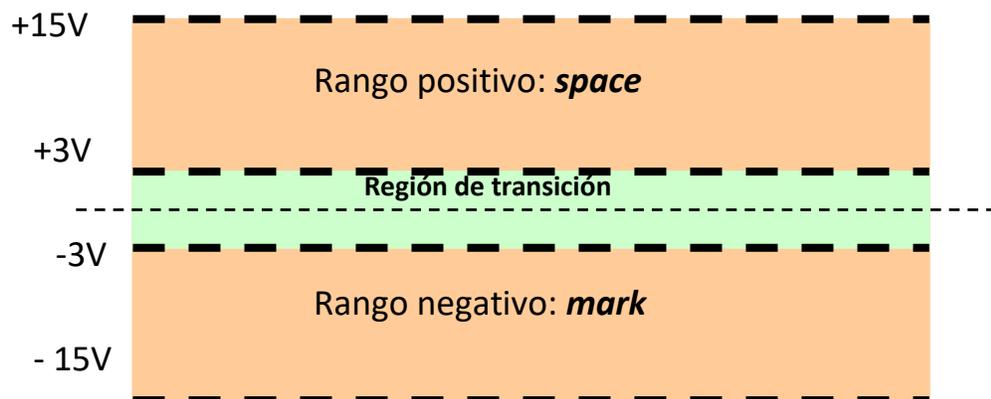
Este Término, viene de los 1ros. integrados con transistores bipolares TTL: Transistor-Transistor- Lógica.

Hoy todas las tecnologías digitales se alimentan con 5V.

- Un uno binario se denomina “marca” (*mark*) y se representa por un voltaje de -3 a -15 voltios.
- Un cero binario se denomina espacio (*space*) y se representa por un voltaje de +3 a +15 voltios.
- Cualquier voltaje entre -3 y +3 voltios se considera inválido.

Una corriente de corto circuito no puede exceder los 500mA

El objetivo de RS-232C que es antigua la norma como los transistores, es aumentar el nivel de tensión a través de la línea (cable) y cuando se recibe volverlo al nivel original TTL, esto se realiza para aumentar la inmunidad al ruido de la señal a transmitir.



A continuación mostramos dos ejemplos:

En el primero se envía desde la PC hacia un receptor el carácter “3” cuyo código ASCII es en hexadecimal el Nro 33. (00110011 en binario)

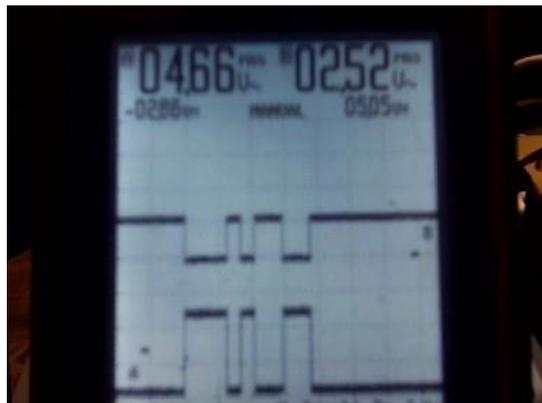
En la figura el trazo superior corresponde a la señal con nivel TTL (0-5 Vcc). Un “0” es Vcc y un “1” es 5 Vcc. De izquierda a derecha el primer bit en bajo es el Bit de START, a continuación están los ocho bit’s de datos empezando desde el menos significativo y a continuación el bit de STOP (“1”)

La traza de abajo muestra la misma información en Norma RS232.

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232</b><br><b>B) Conversión analógica / digital - P.W.M.</b><br><b>Aplicaciones con Arduino</b> |

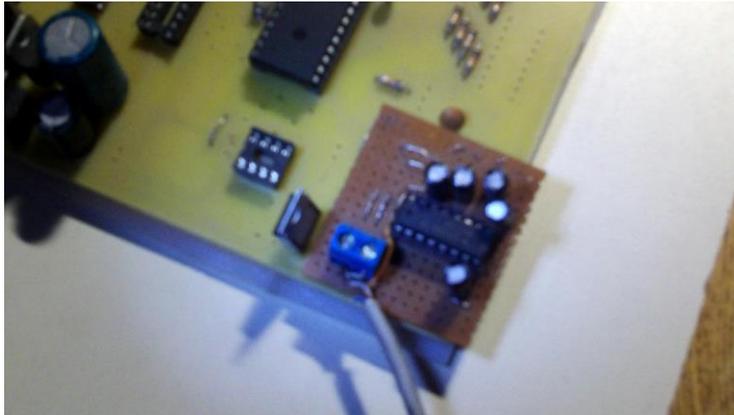


En la figura que sigue se envía el carácter “4” en hexadecimal 34, en binario 00110100. Se utiliza el mismo formato y descripción de la figura anterior



En la siguiente figura se observa el integrado Max 232 con los capacitores externos que se deben agregar. El mismo realiza la conversión TTL- RS232 y viceversa.

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232<br/>B) Conversión analógica / digital - P.W.M.<br/>Aplicaciones con Arduino</b> |



## CONVERSIÓN ANALÓGICA A DIGITAL

### INTRODUCCION

Si bien la mayoría de los equipos electrónicos modernos están controlados por lógica o controladores digitales, vivimos en un mundo análogo así que se hace necesaria la existencia de un dispositivo que convierta los valores que observamos a nuestro alrededor en datos digitales que se puedan analizar.

El convertidor Análogo-Digital (ADC por sus siglas en inglés) es un dispositivo que toma una señal análoga (corriente, voltaje, temperatura, presión, etc.), cuantifica la señal y le asigna un valor que se muestra a su salida en formato digital.

Los ADC más comunes son aquellos que convierten señales de voltaje o corriente a su equivalente digital aunque podemos encontrarlos para medir diversidad de magnitudes análogas. Afortunadamente los Arduino incluyen de fábrica uno o varios ADC que nos permiten convertir niveles de voltaje a valores digitales.

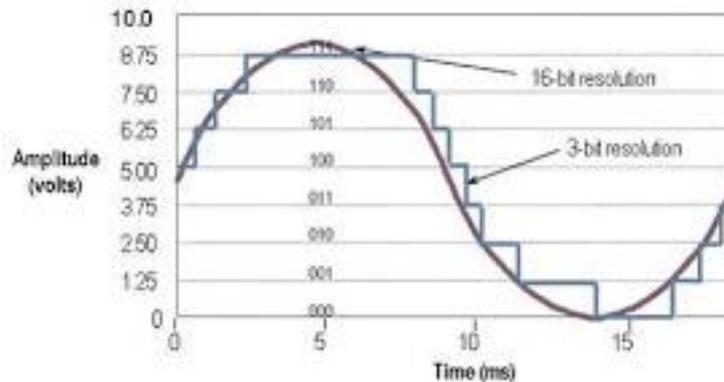
En general hay tres cosas que nos interesa saber sobre un ADC:

1. La resolución
2. El tiempo de conversión
3. El rango de trabajo

Vamos ahora a revisar en detalle cada una de estas tres características.

### La resolución del convertidor

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232<br/>B) Conversión analógica / digital - P.W.M.<br/>Aplicaciones con Arduino</b> |



Uno de los mayores inconvenientes de convertir señales análogas a su equivalente en digital es que pasamos de una señal que por naturaleza es continua y sin cortes a un dato discreto, que solo puede tener ciertos valores. Este proceso es conocido como "cuantificación".

Generalmente la cantidad de valores discretos en los que un ADC puede traducir una señal análoga a digital se conoce como su resolución y usualmente es medida en base al número de bits que el ADC tiene a su salida. El número de bits representa la cantidad máxima de valores discretos o de pasos que un ADC puede tener.

Supongamos por ejemplo que tenemos un convertidor ADC de 3 bits. Este convertidor podrá tener solo 8 valores diferentes a su salida.

### **El tiempo de conversión y la frecuencia de muestreo**

Desafortunadamente la conversión de una señal análoga a su equivalente valor digital no es instantánea, esto significa que el convertidor tarda algún tiempo (usualmente muy pequeño) en realizar el proceso de conversión. Esto limita el número de conversiones que podemos realizar en una unidad de tiempo, a este número de conversiones que puede realizar un ADC usualmente se le conoce como frecuencia de muestreo o en inglés "sampling rate".

Podemos entender este parámetro como la precisión del convertidor en el tiempo. Un ADC con un tiempo de conversión de 1 segundo es más lento que uno que tarde 0.001segundos en hacer una conversión. La frecuencia de muestreo máxima del ADC se calcula utilizando el inverso del tiempo de conversión.

Al igual que el parámetro de la resolución, que la frecuencia de muestreo sea buena o mala depende mucho de que tan rápido cambie el valor análogo que deseamos medir. Por ejemplo si tenemos una sonda que mide la humedad en el suelo, el valor a medir no va a cambiar hasta que transcurran algunos minutos por lo que un ADC que tarde 1 segundo en realizar la conversión no nos va a dar mayor problema, por el contrario si estamos queriendo convertir a digital una señal de audio, 1 segundo de tiempo de conversión se queda sumamente corto para capturar las frecuencias que el oído puede distinguir.

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232<br/>B) Conversión analógica / digital - P.W.M.<br/>Aplicaciones con Arduino</b> |

Una forma simple para elegir un ADC es que la frecuencia de muestreo sea "al menos" el doble de la frecuencia de la señal que queremos convertir ( [Teorema de muestreo de Nyquist-Shannon](#)). Por ejemplo si queremos medir una señal que oscila a 60Hz lo menos que necesitaremos es un ADC que logre tomar 120 muestras por segundo.

### El rango de trabajo

La tercer característica, pero no la menos importante, que debemos tomar en consideración al elegir un ADC es el rango de voltajes/corrientes que acepta en su entrada, este parámetro es esencial para evitar "quemar" (literalmente) nuestro ADC.

La única forma de saber el rango del ADC es revisando su hoja técnica, usualmente encontraremos el rango de valores de voltaje y de corriente aceptados en la entrada como también las tolerancias aceptadas. Para el Arduino Mega, según las especificaciones técnicas el rango de entrada va desde 0v hasta Vcc (5V).

### Voltajes de referencia

El voltaje de referencia es un voltaje utilizado como valor "máximo" por el ADC para realizar el proceso de conversión.

El Arduino Mega puede utilizar dos valores como referencia para realizar la conversión, el primero el voltaje de alimentación (5V) y el otro es un voltaje externo en tanto este no supere el voltaje de alimentación (Vcc).

¿Para que usar un voltaje de referencia externo? Si recuerdan, la resolución de nuestro ADC está limitada a un número de pasos predefinido por el fabricante. En el caso de que la señal que deseamos medir tenga un valor menor al voltaje de alimentación entonces el rango de valores que podemos utilizar se reduce y por lo tanto reducimos también la precisión.

Por ejemplo: Nuestra señal varía desde 0 a 2.5V. Si calculamos nuestro tamaño de "paso" para cada valor utilizando la referencia de la alimentación (5V):

$$\frac{5V}{1024} \approx 0.005 \approx 5mV$$

Pero si en cambio utilizamos el voltaje de referencia de 2.5V:

$$\frac{2.5V}{1024} \approx 0.0024 \approx 2.4mV$$

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232<br/>B) Conversión analógica / digital - P.W.M.<br/>Aplicaciones con Arduino</b> |

La práctica se fundamentará en utilizar el módulo de conversión Analógica Digital del Arduino UNO.

El módulo de conversión utiliza una resolución de 10 bit's o sea, el valor máximo a medir se discriminará en 1024 partes.

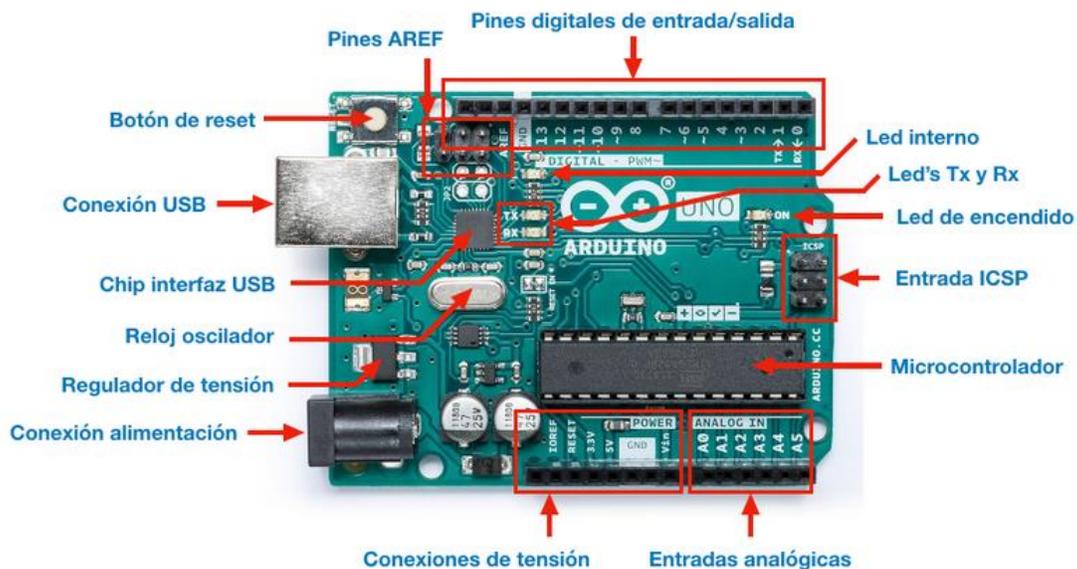
En el caso de un arduino Uno, el valor de 0 voltios analógico es expresado en digital como B0000000000 (0) y el valor de 5V analógico es expresado en digital como B1111111111 (1023). Por lo tanto todo valor analógico intermedio es expresado con un valor entre 0 y 1023, es decir, sumo 1 en binario cada 4,883 mV.

Arduino Uno tiene una resolución de 10 bits, es decir, unos valores entre 0 y 1023.

Arduino Due tiene una resolución de 12 bits, es decir, unos valores entre 0 y 4095.

### Entradas Analógicas en Arduino

Los microcontroladores de Arduino contienen en la placa un convertor analógico a digital de 6 canales. El convertor tiene una resolución de 10 bits, devolviendo enteros entre 0 y 1023. Los pines analógicos de Arduino también tienen todas las funcionalidades de los pines digitales. Por lo tanto, si necesitamos más pines digitales podemos usar los pines analógicos. La nomenclatura para los pines analógicos es A0, A1, etc....



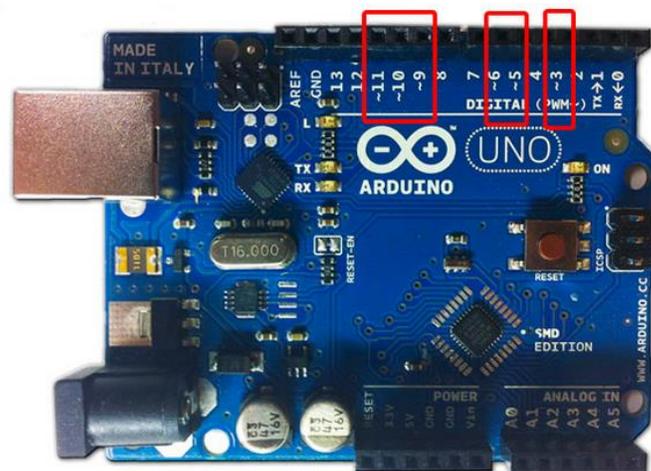
|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232<br/>B) Conversión analógica / digital - P.W.M.<br/>Aplicaciones con Arduino</b> |

## P.W.M.

### Salidas Analógicas. PWM

Como hemos dicho Arduino Uno tiene entradas analógicas que gracias a los conversores analógico digital puede entender ese valor el microcontrolador, pero no tiene salidas analógicas puras y para solucionar esto, usa la técnica de PWM.

Algunos pines digitales pueden usarse como salidas analógicas PWM



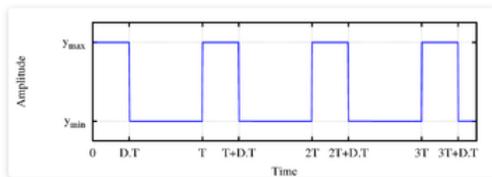
Las Salidas PWM (Pulse Width Modulation) permiten generar salidas analógicas desde pines digitales. **Arduino Uno no posee salidas analógicas puras**, sin embargo el Arduino Due sí tiene salidas analógicas puras mediante dos DAC. El [arduino due](#), posee dos salidas analógicas puras mediante dos conversores digital a analógico. Estos pines pueden usarse para crear salidas de audio usando la [librería correspondiente](#).

La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

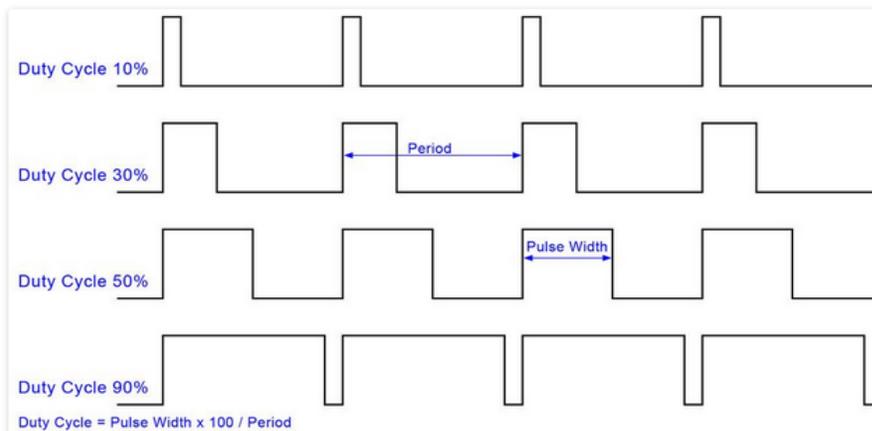
El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período.

duty cycle = (tiempo que la salida está a uno o HIGH)/ (periodo de la función)

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232</b><br><b>B) Conversión analógica / digital - P.W.M.</b><br><b>Aplicaciones con Arduino</b> |

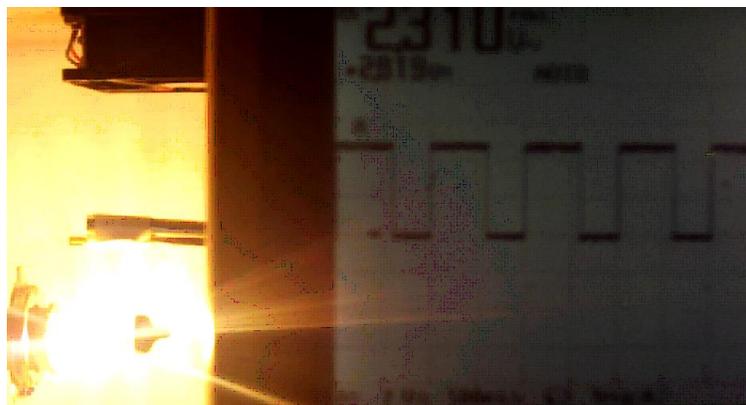
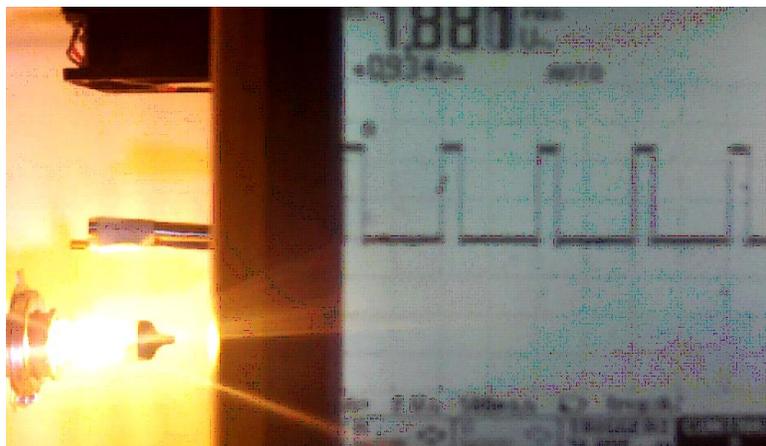


Diferentes valores de una señal PWM:



En las siguientes figuras prestar atención a la relación entre la intensidad de la luz en la lámpara y el tiempo del estado alto de la señal PWM

|  |
|--|
| FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO   |
| ASIGNATURA ELECTRONICA GENERAL Y APLICADA  |
| CARRERA: INGENIERIA INDUSTRIAL   |
| CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS   |
| TP N°10  |
| TITULO: A) Comunicación serie RS 232<br>B) Conversión analógica / digital - P.W.M.<br>Aplicaciones con Arduino |



|  |
|--|
| FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO   |
| ASIGNATURA ELECTRONICA GENERAL Y APLICADA  |
| CARRERA: INGENIERIA INDUSTRIAL   |
| CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS   |
| TP N°10  |
| TITULO: A) Comunicación serie RS 232<br>B) Conversión analógica / digital - P.W.M.<br>Aplicaciones con Arduino |

## CÓDIGO CONVERSIÓN ANALÓGICA/DIGITAL

```

Conversor Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda
Conversor $
const int Led = 13;
const int Pin_A0 = A0;
int valor = 0;
void setup() {
  Serial.begin(9600);
  pinMode(Pin_A0, INPUT);
  pinMode(Led, OUTPUT);
  digitalWrite(Led, LOW);
}

void loop()
{ valor = analogRead(Pin_A0);
  Serial.println(valor);
  if (valor >=650)
  {digitalWrite(Led, HIGH);
  }
  else {digitalWrite(Led, LOW);
  }
  delay(300);
}

```

Definimos una variable “valor” donde guardaremos el valor resultante de la lectura del puerto analógico A0 convertido a digital.

Inicializamos el puerto de comunicaciones con una velocidad de 9600 baudios

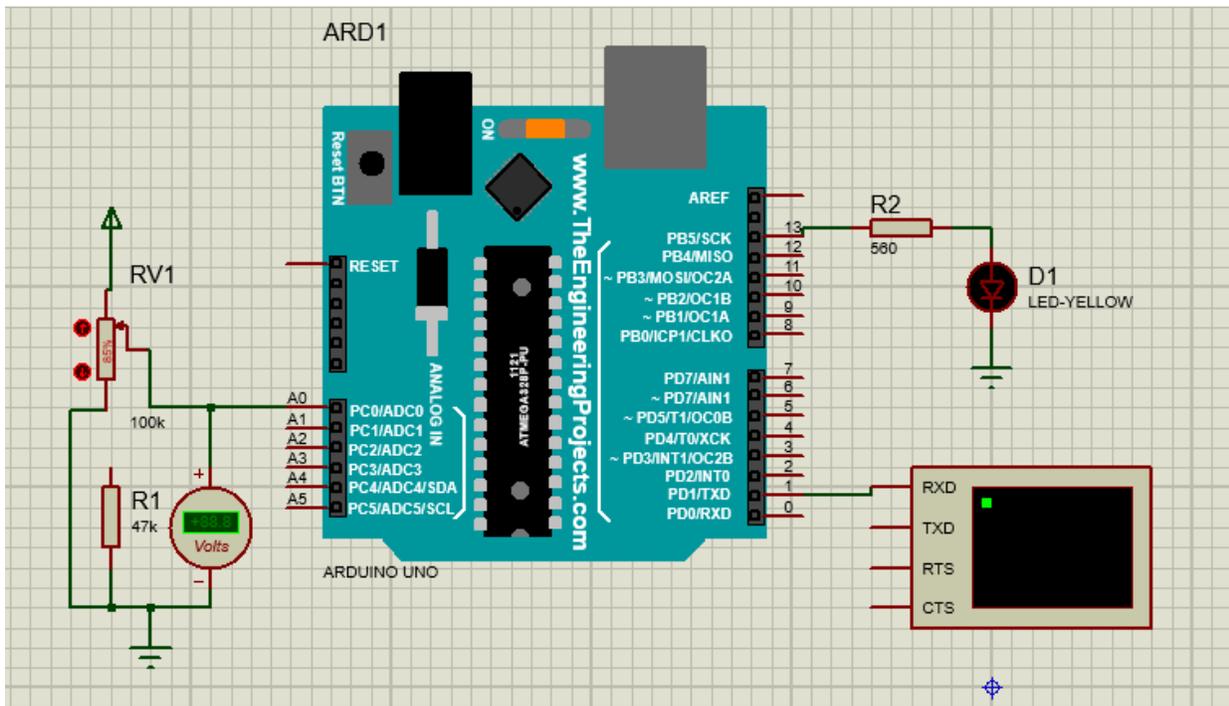
Definimos una entrada analógica (A0) y una salida digital (13)

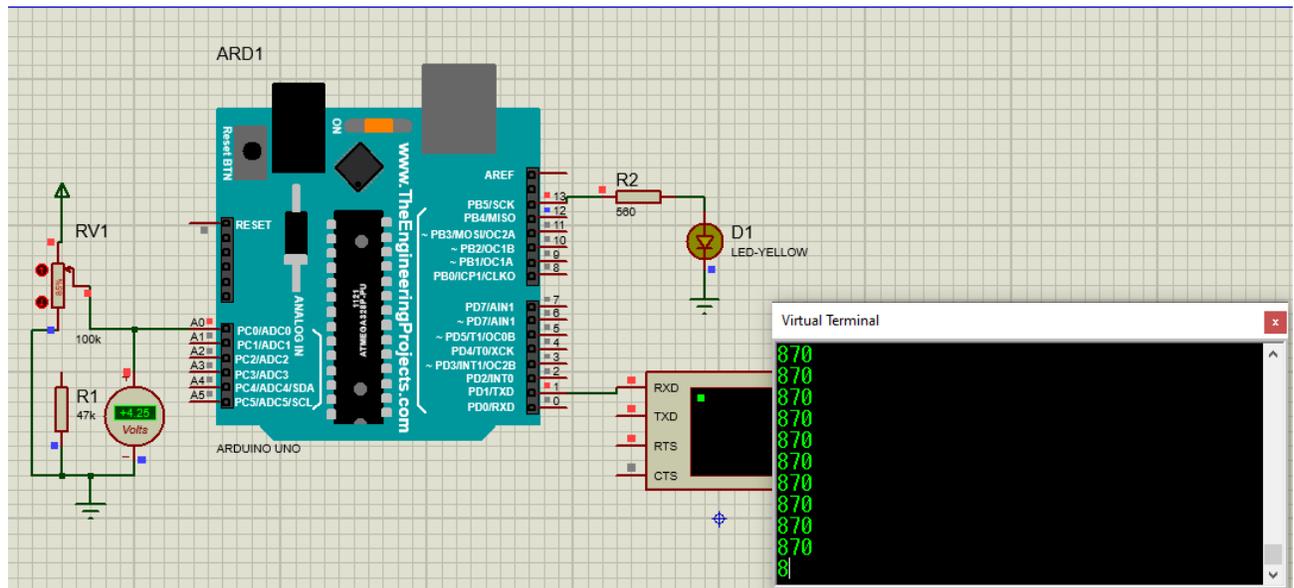
En el lazo se lee la entrada analógica/A0), se la convierte a digital se la envía a la Terminal Virtual, se la compara con el valor 650, si es mayor o igual se enciende el LED conectado a 13, si es menor se apaga el LED

|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232</b><br><b>B) Conversión analógica / digital - P.W.M.</b><br><b>Aplicaciones con Arduino</b> |

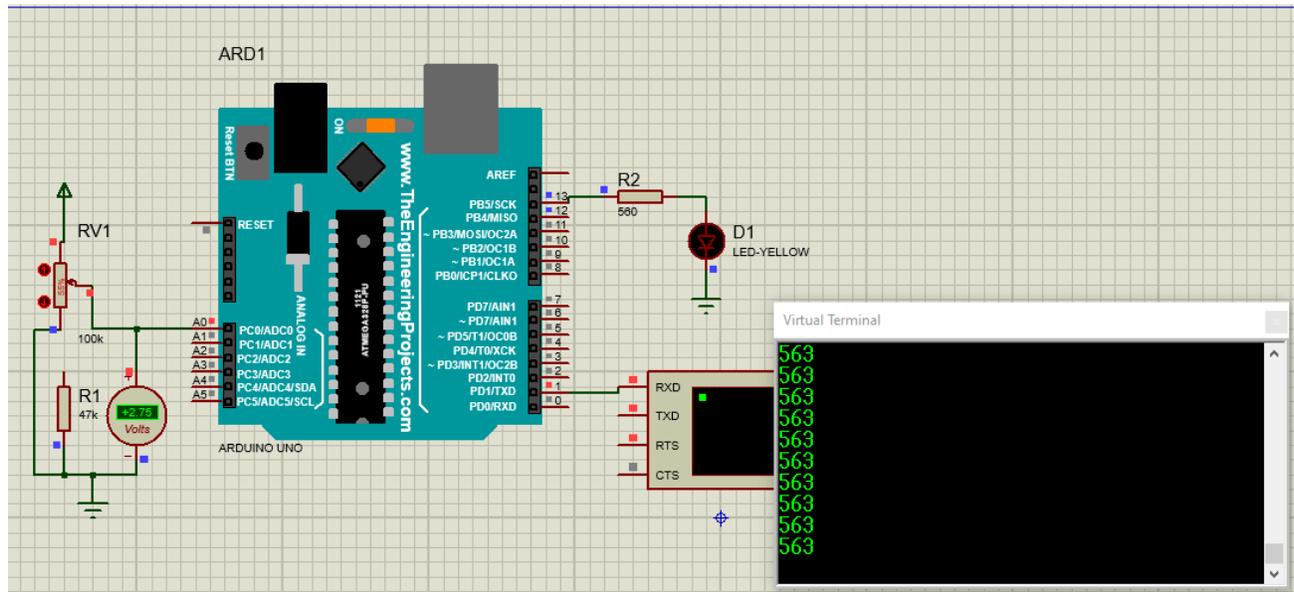
La simulación la realizamos en PROTEUS.

Con un potenciómetro conectado su punto medio a la entrada A0, generamos la señal analógica. Para visualizar el valor leído y convertido, utilizamos la Terminal Virtual





En la imagen superior el valor convertido es 870 (mayor a 650) el LED está encendido.  
 En la imagen inferior el valor convertido 563 (menor a 650) , el LED está apagado



|  |
|--|
| FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO   |
| ASIGNATURA ELECTRONICA GENERAL Y APLICADA  |
| CARRERA: INGENIERIA INDUSTRIAL   |
| CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS   |
| TP N°10  |
| TITULO: A) Comunicación serie RS 232<br>B) Conversión analógica / digital - P.W.M.<br>Aplicaciones con Arduino |

### CÓDIGO RECEPCIÓN SERIE

```

Serie Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda
Serie §
const int Led = 13;
const int Pin_A0 = A0;
int valor = 0;
void setup() {
  Serial.begin(9600);
  pinMode(Pin_A0, INPUT);
  pinMode(Led, OUTPUT);
  digitalWrite(Led, LOW);
}

void loop()
{ valor = analogRead(Pin_A0);
  if (valor >=650)
  {digitalWrite(Led, HIGH);
   Serial.write('E');
  }
  else {digitalWrite(Led, LOW);
   Serial.write('A');
  }
  delay(300);
}

```

Definimos :Led conectado al pin13, Pin\_A0 conectado al pin A0 (entrada analógica).  
 Una variable “valor” donde se guardará el valor convertido leído en A0  
 Inicializamos el puerto de comunicaciones a una velocidad de 9600 baudios  
 Definimos Pin\_A0 como entrada  
 Definimos LED como salida Inicialmente LED apagado.  
 En el lazo se lee, convierte la entrada analógica A0 y se guarda en valor.

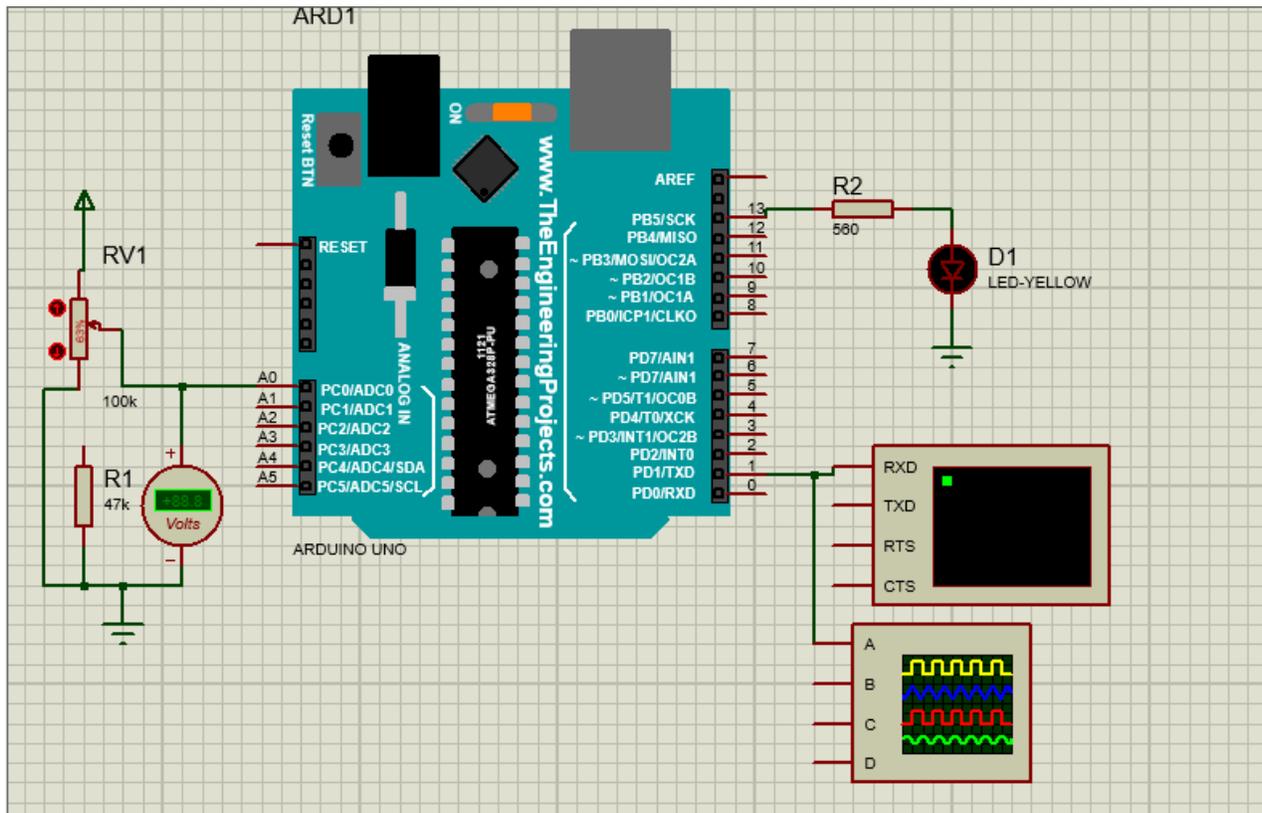
|   |
|---|
| <b>FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO</b>   |
| <b>ASIGNATURA ELECTRONICA GENERAL Y APLICADA</b>  |
| <b>CARRERA: INGENIERIA INDUSTRIAL</b>   |
| <b>CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS</b>   |
| <b>TP N°10</b>  |
| <b>TITULO: A) Comunicación serie RS 232</b><br><b>B) Conversión analógica / digital - P.W.M.</b><br><b>Aplicaciones con Arduino</b> |

Si el valor es mayor o igual a 650, se enciende el LED y se envía por el puerto serie la letra "E"  
Si el valor es menor de 650, se apaga el LED y se envía por el puerto serie la letra "A"

La simulación la realizamos en PROTEUS.

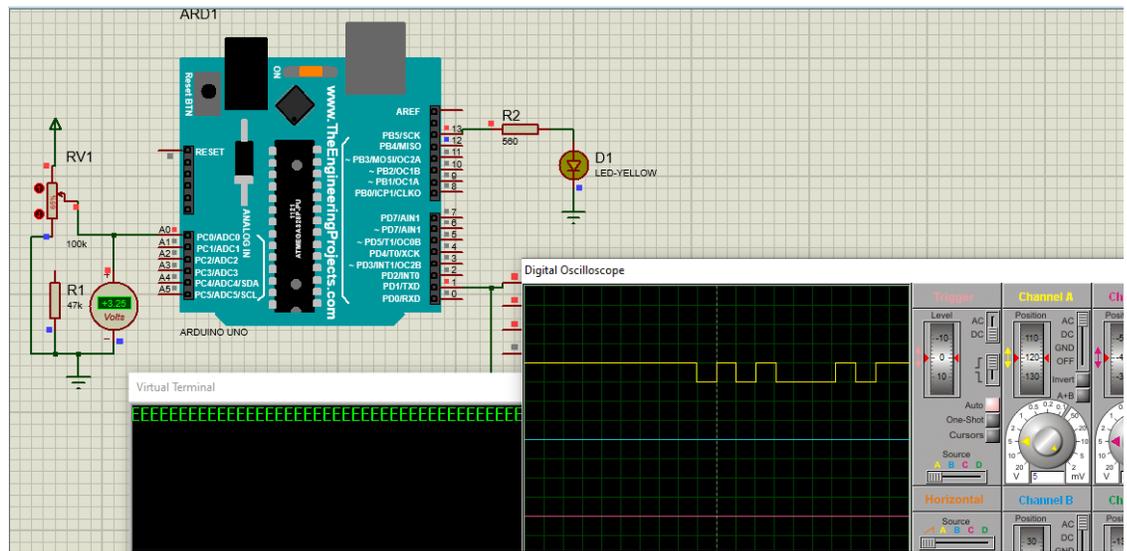
Con un potenciómetro conectado su punto medio a la entrada A0, generamos la señal analógica  
Para visualizar la letra enviada, utilizamos la Terminal Virtual.

Se agrega además un osciloscopio conectado al pin TX para visualizar el código ASCII de la letra enviada.



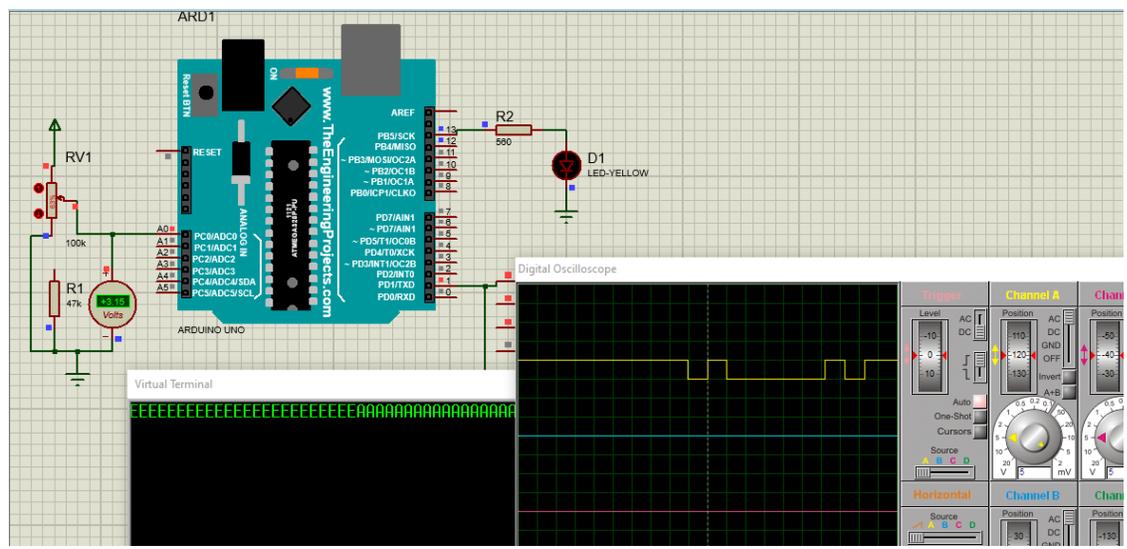
**FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO**  
**ASIGNATURA ELECTRONICA GENERAL Y APLICADA**  
**CARRERA: INGENIERIA INDUSTRIAL**  
**CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS**  
**TP N°10**

**TITULO: A) Comunicación serie RS 232**  
**B) Conversión analógica / digital - P.W.M.**  
**Aplicaciones con Arduino**



En la imagen superior el valor convertido es mayor a 650, el LED está encendido y se envía la letra E. En el osciloscopio se visualiza el ASCII 45 en hexadecimal.

En la imagen inferior el valor convertido es menor a 650, el LED está apagado y se envía la letra A. En el osciloscopio se visualiza el ASCII 41 en hexadecimal.



|  |
|--|
| FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO   |
| ASIGNATURA ELECTRONICA GENERAL Y APLICADA  |
| CARRERA: INGENIERIA INDUSTRIAL   |
| CICLO 2023 - CARPETA DE TRABAJOS PRACTICOS   |
| TP N°10  |
| TITULO: A) Comunicación serie RS 232<br>B) Conversión analógica / digital - P.W.M.<br>Aplicaciones con Arduino |

### TRABAJO PRÁCTICO A DESARROLLAR

#### **Punto A**

Si se transmite un dato de 7 bit's: 1001001 y se está usando para la comunicación en serie paridad par. Cuánto valdrá el bit de paridad

#### **Punto B**

En el siguiente código a qué velocidad se realiza la comunicación. Si la tensión de referencia es 5 Volt, y la conversión digital arroja el valor binario  $1010010111_2$ , el LED (13), está encendido o apagado?

```

conver1 $

int valor = 0 ;
char option = ' ';
void setup() {
  pinMode(13,OUTPUT);
  Serial.begin(2400);
}

void loop()
{
  valor = analogRead(0);
  if (valor >= 657)
  { digitalWrite(13,HIGH);
    Serial.write('e');
  }
  else {digitalWrite(13,LOW);
    Serial.write('a');
  }
  Serial.println(valor);
  delay(200);
}

```

### **RESUMEN DE LA ACTIVIDAD**

**Realice todos los ejercicios indicados en los Puntos A y B.**

**Presente un informe grupal con los resultados.**

**Indique en cada hoja del informe el mismo encabezado que el utilizado en este Trabajo. En el pie de página indique los nombres completos del grupo de trabajo.**