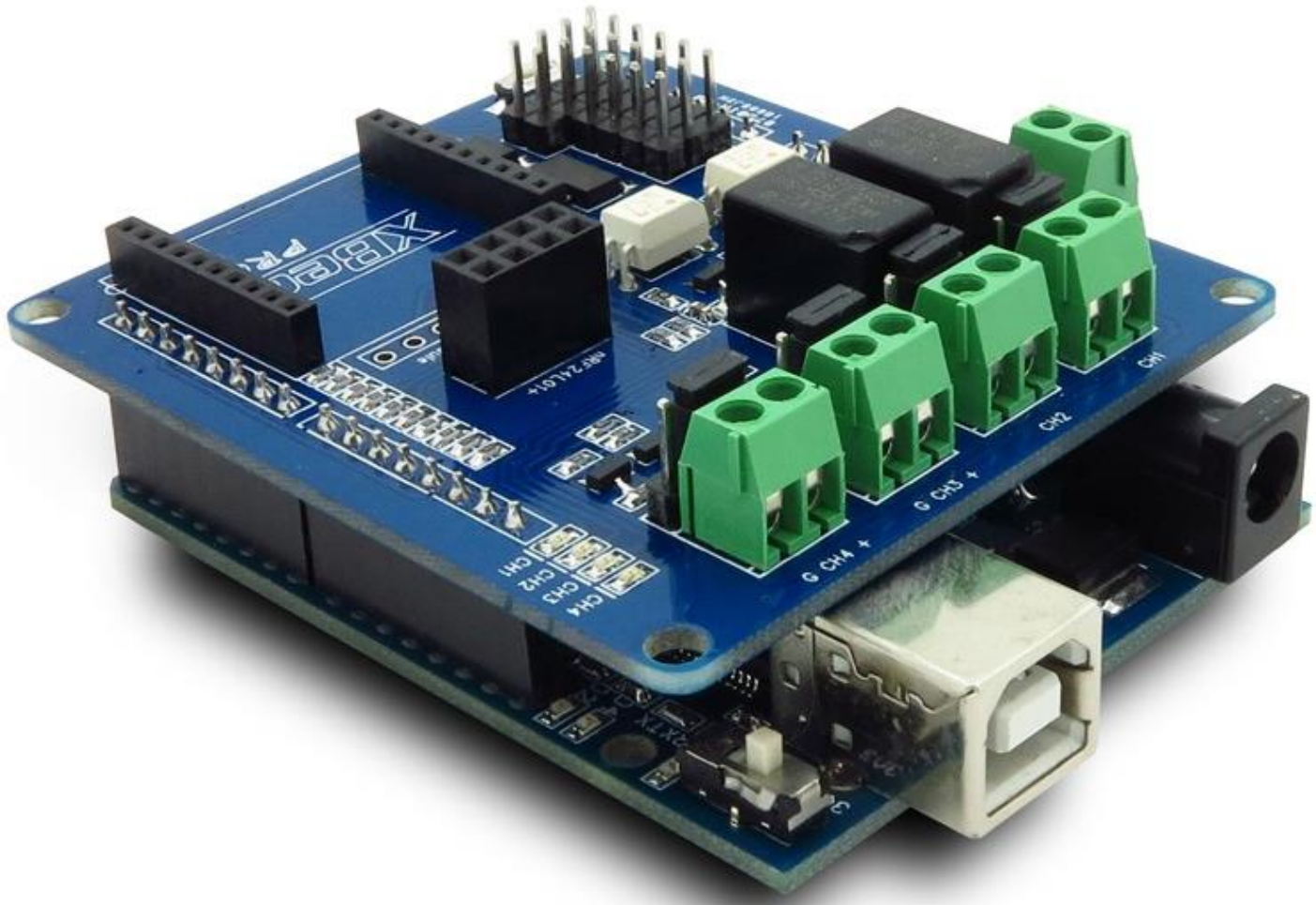

MÓDULOS DE COMUNICACIÓN

SISTEMAS EMBEBIDOS

Los datos transmitidos por el puerto serial pueden ser sometidos a un formato de transmisión. Mas información en ***Arduino Language Reference.***

Serial.begin()
Serial.print()
Serial.println()

Serial.begin()

Rutina para abrir el puerto serial. Junto a la apertura del puerto se debe de definir la velocidad de transmisión de datos.

Sintaxis:

Serial.begin(SPEED)

SPEED: Velocidad en bits por segundo.

Serial.print()

Esta rutina da inicio a la transmisión de caracteres a través del puerto serial.

Sintaxis:

Serial.print(VALOR)

VALOR: cualquier tipo de dato, desde cadenas hasta caracteres.

Serial.println()

Exhibe el mismo comportamiento que Serial.print(), pero a diferencia de este, Serial.println() agregar una salto de línea después de la cadena transmitida.

Serial.print(32, HEX)

Código blinkSerial.ino

```
int led = 13;

void setup() {
        pinMode(led, OUTPUT);
        Serial.begin(9600);
}

void loop() {
        Serial.print("ALTO\n");
        digitalWrite(led, HIGH);
        delay(1000);
        Serial.println("BAJO");
        digitalWrite(led, LOW);
        delay(1000);
}
```

Inicializa el puerto serial con una velocidad de transmisión de 9600 bits por segundo.

IMPORTANTE: El receptor de la comunicación serial debe estar configurado a la misma velocidad de transmisión, de lo contrario la transmisión de información no se llevara a cabo de manera satisfactoria.

El uso de Serial.println() nos ahorra la necesidad de incluir el carácter de retorno de línea (“\n”) en cada cadena enviada.

Entrada
Comunicación Serial

Se puede intercambiar información entre otros dispositivos, permitiendo comunicarnos con la tarjeta de manera mas amigable.

Este puerto es Full-Duplex, por lo que puede enviar y recibir información al mismo tiempo, comportándose entonces como una terminal de Entrada/Salida.

Serial.begin()
Serial.available()
Serial.read()

Serial.begin()

Inicializa el puerto serial de Galileo y define el «**baud rate**» (Velocidad de transmisión de datos).

Sintaxis:

Serial.begin(speed)

speed -> Baud Rate

Serial.available()

Esta función devuelve el numero de bytes disponibles para ser leidos en el puerto serial.

Sintaxis:

Serial.available()

Serial.read()

Lee los datos que entran en el puerto serial.

Regresa -1 si no hay datos disponibles.

Sintaxis:

Serial.read()

Código SerialInput.ino

Variable que recibirá el byte de entrada en el puerto serial

```
int ledPin = 13;
char inByte = 0;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
}

void loop() {
  if (Serial.available() > 0) {
    inByte = Serial.read();
    if (inByte == 'a') {
      if(digitalRead(ledPin)){
        digitalWrite(ledPin, LOW);
      }
      else {
        digitalWrite(ledPin, HIGH);
      }
    }
  }
}
```

Inicializa el puerto serial con una velocidad de transmisión de 9600 bits por segundo.

La terminal 13 se configura como salida.

Se comprueba que existan datos disponibles para su lectura en el puerto serial

Lectura del dato entrante en el puerto serial

Encendido y apagado del LED.

IMPORTANTE: El transmisor de la comunicación serial debe estar configurado a la misma velocidad del receptor, de lo contrario la transmisión de información no se llevará a cabo de manera satisfactoria.

PROGRAMACIÓN

CÓDIGO 9: ENVÍO Y RECEPCIÓN DE MENSAJES

CÓDIGO 10: USO DE VECTORES Y DATOS TIPO STRING

Teclado Matricial

Estos teclados matriciales están conformados por filas y columnas, un teclado matricial de 4×4 tiene 4 filas y 4 columnas interconectadas entre si y si cada intersección representaría un pulsador se tendría 16 pulsadores interconectados entre sí.

Código Keypad.ino

```
#include <Keypad.h>

const byte ROWS = 4; //FILAS
const byte COLS = 4; //COLUMNAS
char keys[ROWS][COLS] = {
  {'7','8','9','*'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'#','0','=','+'}
};
byte rowPins[ROWS] = {5, 4, 3, 2};
byte colPins[COLS] = {9,8, 7, 6};
Keypad keypad = Keypad( makeKeymap(keys), rowPins,
colPins, ROWS, COLS );

void setup() {
  Serial.begin(9600);
}

void loop() {
  char key = keypad.getKey();
  if(key)
  {
   Serial.println(key);
    }
}
```

Libreria de keypad

Inicializar filas y columnas

Creación de la matriz

Orientación de pines y llamada de librería

Condición si tecla ha sido presionada

PROGAMACIÓN

CÓDIGO 11: INGRESO Y VALÍDACIÓN DE CONTRASEÑA POR TECLADO