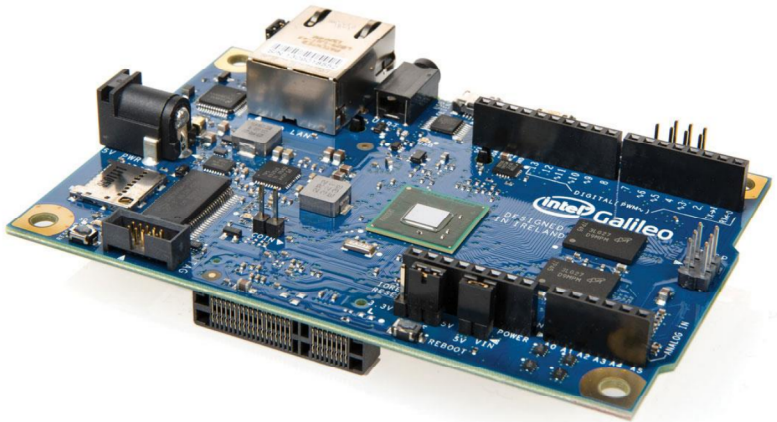
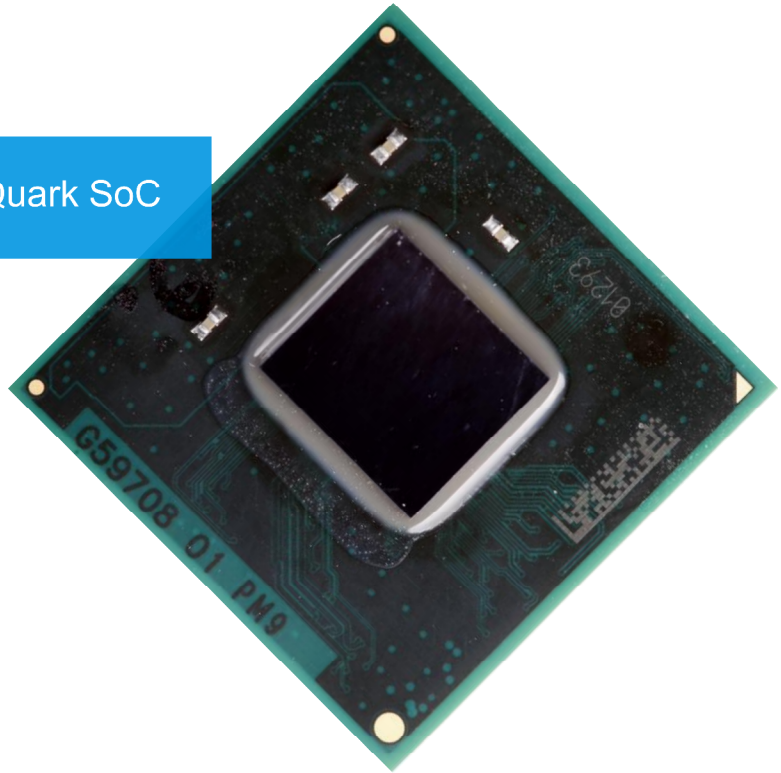


Intel Galileo: Medios de Entrada

Intel Galileo: Medios de Entrada

Intel Quark SoC



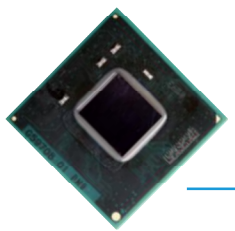
- Medios de Entrada en Galileo
- Entradas Digitales
- Entradas Analógicas
- Entrada de Comunicación Serial
- Aplicación: Joystick
- Aplicación: Teclado Analógico

Medios Entrada

Entradas de Galileo:

La tarjeta Galileo nos proporciona 3 tipos de entrada de datos, con los cuales podemos ingresar información para poder controlar algún periférico o realizar alguna tarea.

1. Entradas digitales de propósito general.
2. Entradas analógicas (Convertidor Analógico-Digital)
3. Entrada Serial con protocolos estándar como: UART, I2C o SPI

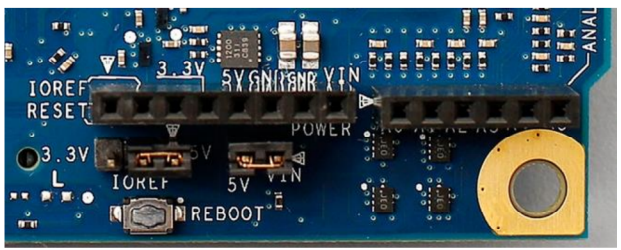


Entradas Digitales de Propósito General

Galileo tiene 14 terminales de entrada digital de propósito general (mismos pines que las salidas) representadas por un identificador numérico.

Este identificar es utilizado durante la programación del dispositivo para interactuar con estos pines.

Mediante el jumper **IOREF** se puede seleccionar que los pines de salida proporcionen voltajes de 5V o



Bibliotecas Arduino

Algunas funciones a utilizar para el uso de las entradas digitales son:

- **pinMode()**
- **digitalRead()**

Para usar las funciones anteriores no es necesario incluir librería alguna.

pinMode()
digitalRead()

pinMode()

Al hacer una llama a la rutina pinMode se le deja saber a Galileo si una terminal será utilizada como entrada o salida.

Sintaxis:

pinMode(pin, MODO)

MODO: INPUT o OUTPUT.

digitalRead()

Después de especificar que una terminal se usará como entrada, se emplea esta función para leer un valor digital alto o bajo presente en dicha terminal.

Sintaxis:

digitalRead(pin)

pin -> Número de terminal a leer

Código
button.ino

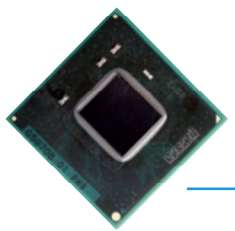
```
const int buttonPin = 2;
const int ledPin =  13;

int buttonState = 0;

void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(buttonPin, INPUT);
}

void loop(){
    buttonState = digitalRead(buttonPin);

    if (buttonState == HIGH) {
        digitalWrite(ledPin, HIGH);
           }
    else {
        digitalWrite(ledPin, LOW);
  }
}
```

Configura la terminal 13 como salida y la terminal 2 como entrada.

Lee el estado de la terminal 2..

Escribe un estado ALTO en la terminal 13

Escribe un estado BAJO en la terminal 13



Entradas Analógicas

La tarjeta Galileo cuenta con 6 entradas analógicas (A0 – A5).

Estas entradas funcionan como convertidores Analógico a Digital (ADCs) **de 10 bits.**

El voltaje de referencia de las entradas analógicas se puede ajustar mediante el pin Vin y el jumper Vin. El jumper Vin es utilizado para configurar un voltaje de referencia de 5V

Cuidado: Si el jumper Vin está conectado y en el pin Vin se conecta otra fuente de poder, podría dañar la tarjeta.

analogRead()

analogRead()

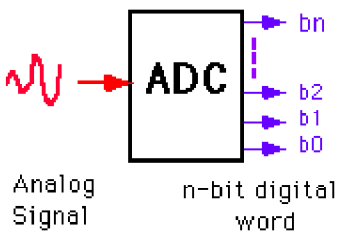
Esta función es utilizada para leer un valor analógico de alguna de las terminales de entrada A0 a A5. Este valor es representado por 10 bits, arrojando valores enteros (**int**) entre 0 y 1023.

Esta función tarda cerca de 100 microsegundos para leer una entrada analógica, por lo que el máximo número de lecturas es de 10,000 por segundo.

Sintaxis:

analogRead(pin)

Donde **pin** es el número de la terminal analógica a leer (A0 – A5)



Código
AnalogInput.ino

```
int sensorPin = A0;
int ledPin = 13;
int sensorValue = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  sensorValue = analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue);
}
```

Se declara una variable que representará la terminal analógica A0.

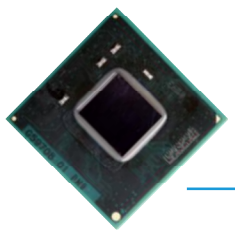
Variable que recibirá el valor analógico representado por 10 bits.

La terminal 13 se configura como salida.

Lectura de la terminal analógica A0.

Se cambia el estado del pin de salida entre ALTO y BAJO.

Retardos que dependen del valor leído de la terminal analógica



Entrada Comunicación Serial

Como se vio anteriormente, la tarjeta Galileo cuenta con un puerto de comunicación serial, con el cual 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;
int 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.

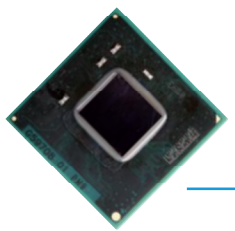
La terminal 13 se configura como salida.

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 llevara a cabo de manera satisfactoria.



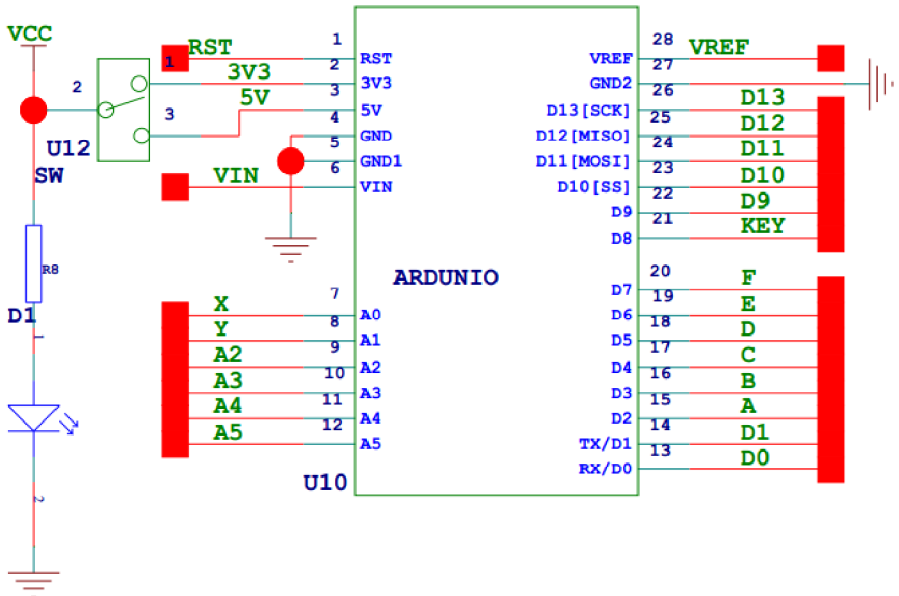
Joystick

Joystick

Un Joystick es una combinación de botones tipo push y un par de potenciómetros.

Para leer la posición de la palanca del Joystick se emplean dos potenciómetros, para el eje X y el eje Y, que se comportan como divisores de voltaje y cuyos valores son leídos por dos terminales de entrada analógica.

Joystick
Esquemático



Código
Joystick_v1.ino

```
const int boton_A=2;

void setup() {
  Serial.begin(9600);
  pinMode(boton_A, INPUT);
}

void loop() {
  Serial.println(digitalRead(boton_A));
  Serial.println(analogRead(A0));
  delay(200);
}
```

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

La terminal 13 se configura como salida.

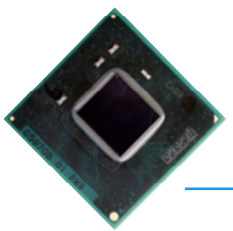
Imprime el estado del botón A

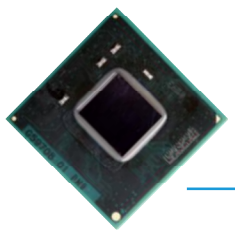
Imprime valor leído de la terminal analógica A0 que corresponde al eje X

Retardo de 200 milisegundos

Reto:

Leer todos los botones y los valores de los potenciómetros y mostrarlos serialmente…



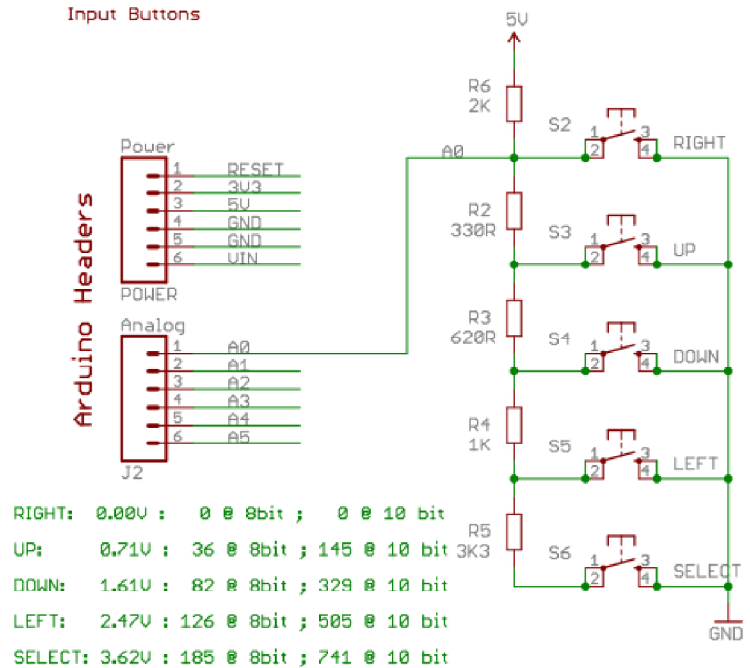


Teclado Analógico

Teclado Analógico

Este tipo de teclados necesitan una sola terminal para leer todos los botones ya que emplean un arreglo resistivo a modo de divisores de voltaje, obteniendo una sola salida. Es necesario que esta salida sea leída por medio de una terminal analógica, la cual detectará el voltaje recibido y por medio de alguna rutina de software se debe determinar la tecla presionada.

Teclado Analógico
Esquemático



Código
AnalogKeypad.ino

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

void loop() {
  if (analogRead(A0) <= 800) {
    delay(10);
    Serial.println(analogRead(A0));
    delay(500);
    }
}
```

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

Se comprueba que se haya presionado un botón

Retardo de 10 segundos para evitar el ruido por rebote del botón tipo push

Imprime valor leído de la terminal analógica A0 que corresponde a uno de los botones

Retardo de 500 milisegundos

Reto:

Imprimir el botón presionado, es decir, si se presiona el botón «SELECT» se mostrará en la terminal la palabra: **SELECT**