

Robot Arduino controlado mediante sensores y con respuestas sobre actuadores

Asier Pérez de Lazarraga

Grado en Tecnologías de Telecomunicación

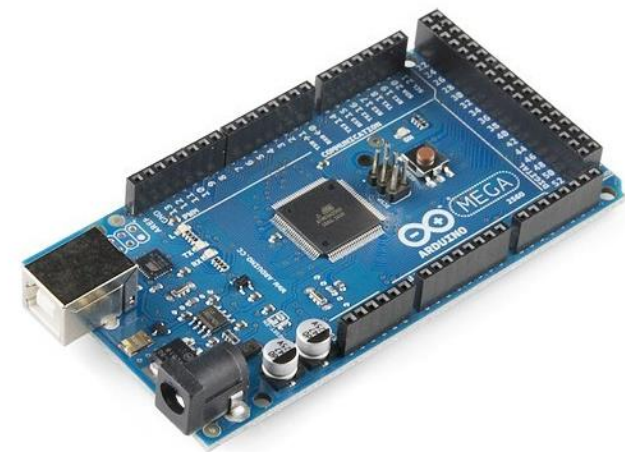
Consultor: José López Vicario

Índice

- ▶ Contexto y Justificación
- ▶ Objetivos del Proyecto
- ▶ Unidad Central: Microcontrolador vs Microprocesador
- ▶ Hardware
- ▶ Comunicación del Robot
- ▶ Sensórica
- ▶ Puertos de expansión
- ▶ Entornos de desarrollo (IDE's)
- ▶ Captura de datos
- ▶ Conclusiones y líneas de trabajo futuras

Contexto y justificación

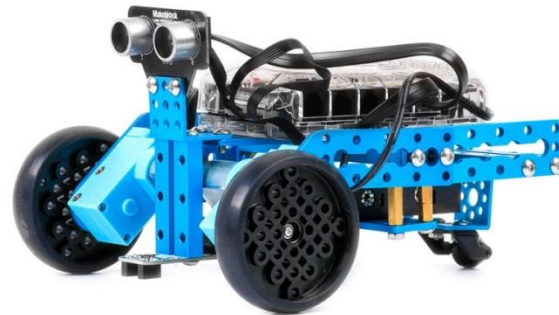
- ▶ Arduino:
 - Hardware libre
 - Entorno de desarrollo (IDE) open-source
- ▶ Contexto:
 - Cultura Do it Yourself (DIY)
 - Auge de cultura contemporánea *maker*
 - Tecnologías IT como base de nuevos conceptos:
 - Smart City
 - Internet of Things (IoT)
 - Industria 4.0
 - Robótica colaborativa
- ▶ Justificación. HW y SW libre para:
 - Emprender
 - Aplicar en entornos propietarios



Arduino MEGA 2560

Objetivos

- ▶ **Objetivos principales:**
 - Construir un robot Arduino con capacidad de desplazamiento
 - Implementar una aplicación para controlar los movimientos del robot de forma pasiva/activa
- ▶ **Otros objetivos:**
 - Equipar de sensores al robot y hacerlo interconectable
 - Posibilitar que el dispositivo sea ampliable



Unidad Central

- ▶ Menor consumo
 - ▶ Más robusto
 - ▶ Orientado a la gestión de entradas y salidas
 - ▶ Similar a un autómata programable (PLC)
- ▶ Mayor capacidad de computación
 - ▶ Mayor velocidad de operaciones
 - ▶ Para múltiples propósitos
 - ▶ Similar a un PC



VS

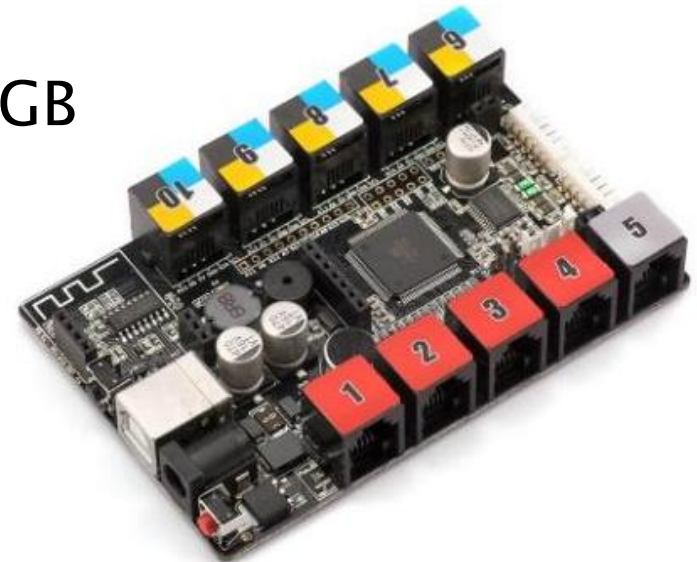


Microcontrolador
Ej: Arduino

Microprocesador
Ej: Raspberry...

Hardware

- ▶ Robot con placa Me Auriga:
 - Basada en microcontrolador Arduino MEGA 2560
- ▶ Microcontrolador (μC): encargado de realizar acciones en función de las entradas recibidas
- ▶ Equipado con:
 - Sensores: luz, temperaturas, sonido, acelerómetro y sensor de giro
 - Actuadores: *buzzer*, 12 LED's RGB
- ▶ Fácilmente ampliable:
 - 10 puertos de expansión RJ25
 - 2 puertos para motor encóder



Comunicación I

Conexión cableada

- ✗ Limita la movilidad
- ✓ No requiere alimentación externa



NFC

- ✗ No válido : rango de alcance limitado (20cm).



Wifi

- ✓ Alta velocidad de datos (no requerida)
- ✗ Alto consumo



ZigBee

- ✓ Alternativa técnicamente válida
- ✗ Dispositivos no compatibles



Comunicación II

- ▶ Solución inalámbrica validada: Bluetooth
 - Distancia de 10 a 15m en campo abierto
 - Consumo de potencia bajo (menor incluso en BLE)
 - Velocidad de transmisión de datos suficiente:
 - No se requieren enviar ni recibir grandes volúmenes de datos
 - Bluetooth clásico: 0,7 – 2,1Mbps
 - BLE: 300kbps
 - Interconexión sencilla
 - Banda ISM de 2,4GHz



Sensórica

- ▶ Dispositivos que detectan una acción externa y la transmiten adecuadamente (según RAE).
- ▶ Existen sensores de todo tipo
- ▶ Por sus salidas pueden ser:
 - Analógicos: mostrar el valor de la temperatura...
 - Digitales: determinar presencia/ausencia...
- ▶ Entradas para múltiples aplicaciones:
 - Automatización industrial
 - IoT
 - ...



Puertos de expansión

- ▶ Disponibles 10 puertos de expansión RJ25:

Puertos	Color	Función	Módulos
1-4	Rojo	V_{CC} 6-12V	Motores
5	Gris	Puerto serie	Comunicación
6-10	Amarillo	Interfaz digital single	Sensores y actuadores
	Azul	Interfaz digital dual	
	Negro	Interfaz analógica	
	Blanco	Puerto I ² C	

- ▶ Fácil expansión
- ▶ Sensores conectables a puertos 6 a 10:
 - Ultrasonidos (10), Sensor IR (9).

IDE I: Entornos Makeblock

- ▶ IDE del propietario con librerías de los robots
- ▶ Versiones:
 - Ordenador (mBlock)
 - Dispositivos móviles (Makeblock)
- ▶ Ejecución del programa *online*
- ▶ Basados en lenguaje *Scratch*:
 - Lenguaje de bloques
 - Lenguaje gráfico
 - Simple



*Panel de control y visualización
Makeblock en móvil Android*

IDE II: Scratch en Makeblock

mBlock - Based On Scratch From the MIT Media Lab(v3.4.8) - Puerto serie Conectado - Guardado

Archivo Editar Conectar Placas Extensiones Lenguaje Ayuda

asier

temperatura a bordo 31.820862
d_actual 58.637932

distancia	
279654	62.844826
279655	58.534481
279656	58.534481
279657	58.655174

Luminosidad: 262

Objetos
Escenario 1 fondo
Fondo nuevo:

M-Panda

Programas Disfraces Sonidos

Movimiento Apariencia Sonido Lápiz Datos y Bloques

Eventos Control Sensores Operadores Robots

Auriga **Conectado**

Programa de Auriga

- avanzar a velocidad 0
- avanzar 1000 grados a la velocidad
- establecer el codificador del motor de
- establecer el codificador del motor de
- establecer el motor DC Puerto1 velocidad
- fijar servo Puerto6 Banco1 ángulo
- fijar motor de pasos motor Puerto1
- establecer el codificador del motor Pu
- fijar display 7-segments Puerto6 número
- fijar led a bordo todos rojo 0 verde
- fijar led Puerto6 posición todos rojo
- fijar tira led Puerto6 Banco2 todos
- reproducir tono en la nota C4 pulsa

Código del programa

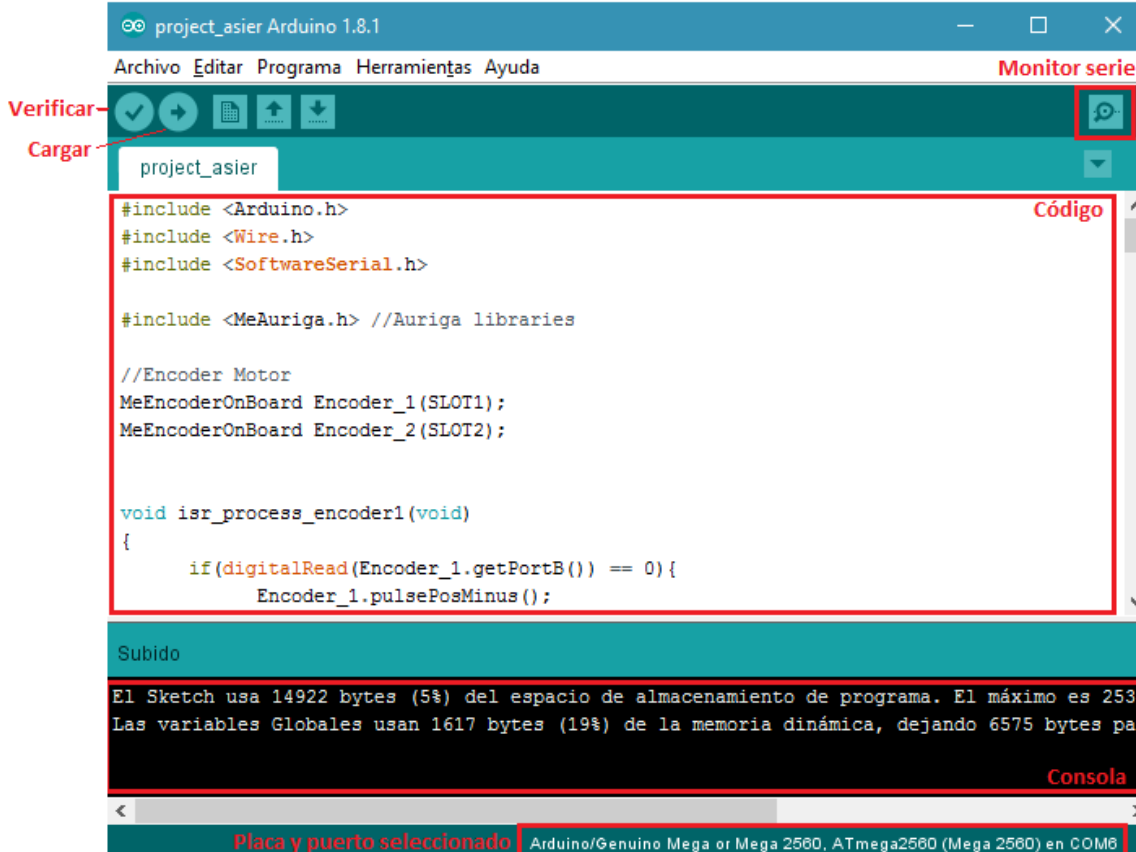
```
al presionar tecla espacio
por siempre
  fijar temperatura a bordo a temperatura a bordo °C
  decir unir Luminosidad: sensor de luz a bordo
  esperar 1 segundos
  si sensor de luz a bordo > 500 entonces
    fijar led a bordo todos rojo 150 verde 150 azul 150
    esperar 0.9 segundos
    fijar led a bordo todos rojo 0 verde 0 azul 0
  si no
    si sensor de luz a bordo > 200 entonces
      fijar led a bordo todos rojo 0 verde 0 azul 150
      esperar 0.9 segundos
      fijar led a bordo todos rojo 0 verde 0 azul 0
    si no
      fijar led a bordo todos rojo 0 verde 200 azul 150
      esperar 0.9 segundos
      fijar led a bordo todos rojo 0 verde 0 azul 0

al presionar tecla flecha arriba
por siempre
  fijar d_actual a distancia del sensor de ultrasonidos Puerto10
  insertar distanciad el sensor de ultrasonidos Puerto10 en último de distancia
  si distanciad el sensor de ultrasonidos Puerto10 > 50 entonces
    avanzar a velocidad 200
  si no
    si distanciad el sensor de ultrasonidos Puerto10 > 25 entonces
      avanzar a velocidad 100
  si no
    si distanciad el sensor de ultrasonidos Puerto10 < 25 entonces
```

Visualización de datos recibidos del robot

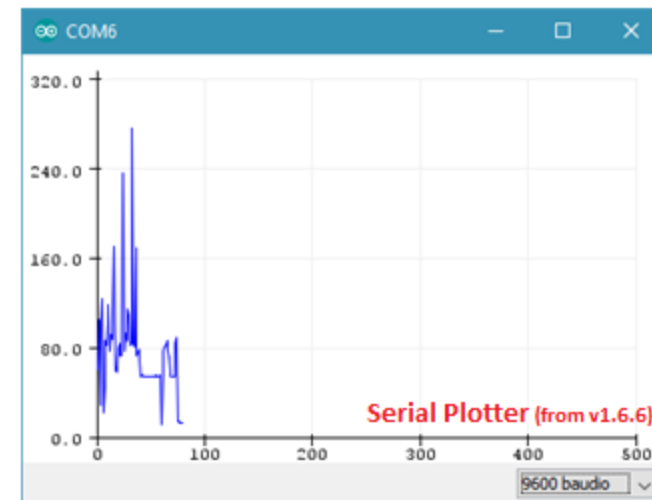
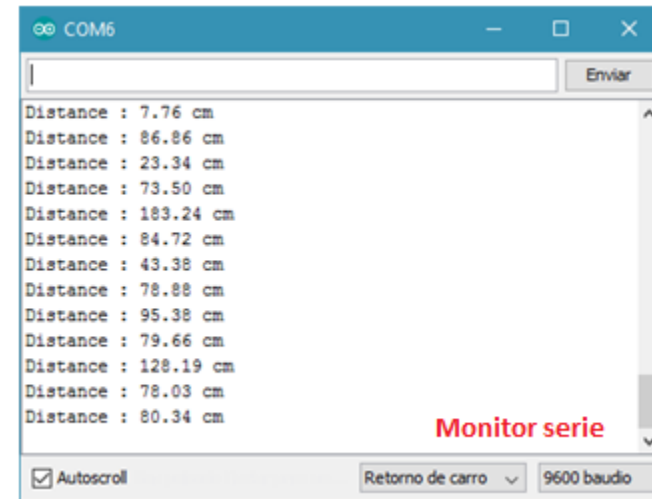
Bloques de código, variables... para usar en el programa

IDE III: Arduino IDE



Arduino IDE

- Lenguaje C/C++
- Se carga en el μ C



Captura de datos

- ▶ Software de captura de puertos conectados (ej: RealTerm):
 - Visualizan los datos
 - Capturan datos → almacenables en ficheros
- ▶ Independencia de puertos con IDE's
- ▶ Posibilidad de procesar los datos de los ficheros que se generen (BBDD, Excel...)
- ▶ Posibilidad de alimentar otros sistemas



Conclusiones

- ▶ Se ha conseguido implementar un robot con capacidad de movimiento
- ▶ Se han implementado programas que producen acciones en función de entradas en:
 - Scratch
 - Arduino (C/C++)
- ▶ Se ha realizado un trabajo de documentación
- ▶ Se ha constatado que existen infinidad de posibilidades a implementar
- ▶ Filosofía DIY / *maker*. Comunidad muy amplia y con muchos recursos y herramientas en red.

Líneas de trabajo futuras

- ▶ Equipar al robot con más sensores/actuadores para cubrir más necesidades
- ▶ Procesamiento y visualización de datos
- ▶ Controlar al robot mediante alguna plataforma de diseño de sistemas (ej.: LabView)
- ▶ Desarrollar y depurar más profundamente el código Arduino

El conocimiento y las herramientas están al alcance y son baratas
La clave es identificar necesidades que requieran ser cubiertas

Gracias por su atención