Maleta de la Innovación 4.0



SmartHome Kit +Componentes extra







Versión del documento: 2

Juanjo López

juanjose.lopez@salesianos.edu arduinoblocks@gmail.com

Índice

Arduino UNO PLUS + Sensor Shield 5	5
Conexiones	6
ArduinoBlocks	7
ArduinoBlocks-Connector v5	7
Parpadeo de un led	8
Salidas PWM	9
Regulación de intensidad de led	10
Escala musical	11
Melodía	13
Melodías RTTTL	15
Led controlado por pulsador	16
Led controlado con pulsador (cambio de estado)	17
Detector de movimiento + Led	19
Led RGB (colores aleatorios)	20
Led RGB (colores fijos)	21
Relé	22
LDR (consola serie)	24
LDR (serial plotter)	26
LDR + Relé (encendido automático por nivel de luz)	29
Control de servo (posicionamiento básico)	30
Control de servo (movimiento suave)	32
Control de servo (osciladores)	35
Motor DC (ventilador)	37
DHT22 (consola / serial plotter)	39
BUS I2C	40
LCD (textos básicos)	41
Termómetro con LCD y DHT22	43
Símbolos personalizados	44
Sensor CO2: CSS811	46
Medidor CO2 con LCD y sensor CCS811	48

Semáforo CO2 con Led RGB y sensor CSS811	50
Mando a distancia IR	52
Piano IR	55
Servo IR	57
Sensor humedad suelo - Riego automático	59
Sensor magnético (Velocímetro bicicleta)	61
Sensor de sonido	64
Funciones	67
Tareas / Multitarea	69
SmartHome: conexionado + programación básica	70
Comunicaciones - Serie / Bluetooth	76
Bluetooth + SmartHome Kit App	87
Bluetooth + AppInventor	91
Otras placas y kits	96

El kit de la "Maleta de la Innovación 4.0" es un kit ofrecido por la Escuela Politécnica de Valencia, Campus de Alcoi dentro de la cátedra SmartCity a los centros educativos de Alcoy.

El kit se basa en el Smart Home Kit de keyestudio, un kit basado en Arduino con múltiples sensores, actuadores y periféricos además de las partes de la maqueta cortadas y listas para ensamblar.

https://shop.innovadidactic.com/es/standard-placas-shields-y-kits/1455-keyestudio-smart-home-paraarduino-con-placa-keyestudio-plus.html

El kit se ha complementado con sensores y módulos extra:

- Sensor de CO2 <u>https://shop.innovadidactic.com/es/standard-sensores/983-keyestudio-ccs811-sensor-de-eco2-dioxido-de-carbono-equivalente-y-tvoc.html</u>
- Sensor de sonido
 https://shop.innovadidactic.com/es/standard-sensores/630-keyestudio-sensor-de-sonido-analogico-con-potenciometro.html
- Sensor y mando IR
 <u>https://shop.innovadidactic.com/es/standard-sensores/668-keyestudio-kit-de-control-remoto-y-r</u>
 <u>eceptor-infrarrojo.html</u>
- Módulo de relé doble
 <u>https://shop.innovadidactic.com/es/standard-actuadores/649-keyestudio-modulo-de-rele-dual-o</u>
 <u>-dos-canales.html</u>
- Sensor magnético (Hall)
 <u>https://shop.innovadidactic.com/es/standard-sensores/615-keyestudio-sensor-de-campo-magn</u>
 <u>etico-hall.html</u>
- Sensor DHT22 (temperatura y humedad) <u>https://shop.innovadidactic.com/es/standard-sensores/1468-keyestudio-sensor-de-temperatura</u> <u>-y-humedad-dht22.html</u>
- Placa solar USB
 <u>https://shop.innovadidactic.com/es/cables/887-placa-solar-con-cable-usb-6v-3-5w-580ma.html</u>
- Batería USB (tipo power bank)
 <u>https://shop.innovadidactic.com/es/otros-steam-y-makers/1558-bateria-auxiliar-portatil-de-220</u>
 <u>0-mah.html</u>

Material compatible con Arduino y ArduinoBlocks (Distribuidor oficial Keyestudio España) <u>https://shop.innovadidactic.com/es/</u>

La programación del kit se realiza de forma gráfica mediante la plataforma ArduinoBlocks <u>http://www.arduinoblocks.com/</u>

En esta guía manual se ha recopilado la información y prácticas realizadas durante la formación.

Arduino UNO PLUS + Sensor Shield 5

La placa Arduino PLUS incluida en el kit es un clon Arduino 100% compatible con algunas mejoras y además incorpora una "sensor shield" para facilitar la conectivida de forma modular.



https://wiki.keyestudio.com/KS0486_Keyestudio_PLUS_Development_Board_(Black_And_Eco-friendl v)

Sensor Shield 5



https://wiki.keyestudio.com/Ks0004_keyestudio_Sensor_Shield_V5

Conexiones

Esquema general de conexiones del kit completo (utilizado en los ejemplos por separado también)



Programación con ArduinoBlocks

ArduinoBlocks es un entorno visual de programación basado en bloques, desarrollado por Juanjo López. Es un entorno web, que además de permitir la creación de programas permite gestionar proyectos de forma integral añadiendo documentación, comentarios, adjuntos, etc.

ArduiniBlocks incorpora funcionalidades específicas para gestión de alumnos y proyectos por parte de los profesores. Además soporta las placas más utilizadas en los entornos educativos.

ArduinoBlocks cuenta con el apoyo de la empresa InnovaDidactic y garantiza la compatibilidad con sus productos de Keyestudio para Arduino (InnovaDidactic es el distribuidor oficial de keyestudio en España)

http://www.arduinoblocks.com/

Todos los colaboradores o entusiastas de ArduinoBlocks colaboran activamente realizando documentación y guías prácticas para docentes que están totalmente disponibles de forma gratuita:

http://www.arduinoblocks.com/web/site/doc

El libro escrito por Juanjo López sobre programación con ArduinoBlocks, actualmente está disponible de forma totalmente libre para descarga en su versión "free book" online y se va actualizando continuamente.

https://www.amazon.es/ArduinoBlocks-edici%C3%B3n-Programaci%C3%B3n-Bloques-Arduino/dp/19 77676588

https://docs.google.com/document/u/1/d/e/2PACX-1vQSrOKHpbLQHVbGFdAvp7DcndoftoHDI20nvw GMaxu_7bGc1bUCmi4U6DZrJWRSudc2iXBg43QMuzCT/pub

ArduinoBlocks-Connector v5

Para poder subir y compilar el programa desde la web de ArduinoBlocks debemos instalar la versión de ArduinoBlocks Connector v5 correspondiente para cada sistema operativo, además de asegurarnos de tener correctamente instalado el driver de Arduino en el sistema.

http://www.arduinoblocks.com/web/site/abconnector5



Parpadeo de un led

Módulo Led:



https://wiki.keyestudio.com/KS0016 Keyestudio White LED Module



Salidas PWM

Arduino no tiene salidas puramente analógicas, pero podemos imitar a una salida analógica mediante la técnica PWM (Pulse Width Modulation = Modulación en Anchura de Pulso).

Ejemplo: Gráficas del funcionamiento del PWM:



La técnica PWM permite tener salidas "pseudo-analógicas" que podrán ser utilizadas para regular la intensidad que se aplica a un actuador, como por ejemplo un led (intensidad), motor (velocidad de giro), servo (posición), etc.

Los pines compatibles con PWM en Arduino UNO son:



3, 5, 6, 9, 10, 11

Y están indicados con el símbolo ~

En ArduinoBlocks podemos escribir en salidas PWM el valor de 0 a 255 con el bloque:



Regulación de intensidad de led

Módulo Led:



https://wiki.keyestudio.com/Ks0234_keyestudio_Yellow_LED_Module



Programa 1: aumento y disminución de intensidad progresiva del led





Programa 2: Aumento de intensidad en saltos

Escala musical

El zumbador pasivo permite emitir tonos en distintas frecuencias, de forma que podemos utilizarlo para emitir señales acústicas, o generar melodías.

Módulo zumbador:



https://wiki.keyestudio.com/Ks0019_keyestudio_Passive_Buzzer_module



<u>Melodía</u>

Módulo zumbador:



https://wiki.keyestudio.com/Ks0019_keyestudio_Passive_Buzzer_module



Programa con melodía de ejemplo:

In	icializar
В	ICIE
	Zumbador Pin 3 Ms (150 Hz (294)
	Esperar 50 milisegundos
	Zumbador Pin 3 Ms (150 Hz (294)
	Esperar 50 milisegundos
	Zumbador Pin 3 Ms (150 Hz (294)
	Esperar 50 milisegundos
	Zumbador Pin 3 Ms (900 Hz (392
	Esperar (150) milisegundos
	Zumbador Pin 3 Ms 900 Hz 587
	Esperar 50 milisegundos
	Zumbador Pin 3 Ms (150 Hz) 523
	Esperar 150 milisegundos
	Zumbador Pin 3 Ms (150 Hz (494)

Esperar (50 milisegundos
Zumbador Pin 3 Ms (150 Hz (440)
Esperar (50 milisegundos
Zumbador Pin 3 Ms (900 Hz (784)
Esperar (150) milisegundos
Zumbador Pin 3 Ms (900) Hz (587)
Esperar (100) milisegundos
Zumbador Pin 3 Ms (150 Hz (523)
Esperar (50 milisegundos
Zumbador Pin 3 Ms (150 Hz (494)
Esperar 50 milisegundos
Zumbador Pin 3 Ms 900 Hz 784
Esperar (150 milisegundos
Zumbador Pin 3 Ms 900 Hz 587
Esperar (100) milisegundos
Zumbador Pin 3 Ms (150 Hz (523)
Esperar 50 milisegundos
Zumbador
Esperar 50 milisegundos
Zumbador Pin 3 Ms (150 Hz (523)
Esperar () 50 milisegundos
Zumbador Pin 3 Ms (1200) Hz (440)

Melodías RTTTL

Módulo zumbador:



https://wiki.keyestudio.com/Ks0019 keyestudio Passive Buzzer module

Esquema de conexiones:



Más información y melodías RTTTL: <u>http://www.arduinoblocks.com/web/help/rtttl</u>

Led controlado por pulsador

Módulo pulsador:



https://wiki.keyestudio.com/Ks0029 keyestudio Digital Push Button

Módulo Led:





Programa de ejemplo:



Led controlado con pulsador (cambio de estado)

Módulo pulsador:



https://wiki.keyestudio.com/Ks0029_keyestudio_Digital_Push_Button

Módulo Led:



Esquema de conexiones:



Programa de ejemplo:



Detector de movimiento + Led

El sensor de movimiento/presencia PIR permite detectar movimiento de cuerpos que emitan radiación infrarrojo (seres vivos)

Módulo detector de movimiento PIR:



https://wiki.keyestudio.com/Ks0052_keyestudio_PIR_Motion_Sensor

Módulo Led:



Programa de ejemplo:



Led RGB (colores aleatorios)

El módulo RGB (de ánodo común) incorpora 3 leds: Rojo, Verde, Azul (con el ánodo todos conectados a VCC, y controlamos la intensidad de cada color mediante PWM de forma inversa, siendo 0v la máxima intensidad)

Módulo led RGB:



https://wiki.keyestudio.com/Ks0032_keyestudio_RGB_LED_Module



Programa de ejemplo:

Bucle								
Establecer rojo • = Establecer verde • = Establecer azul • =	entero aleatorio de (0) entero aleatorio de (0)	a (255) a (255)						
Led RGB	Ánodo v común Pin R 9	• Pin G 10	🔹 Pin B 🕻	11 • R 🕻	rojo 🔹 G	verde •	B	

Led RGB (colores fijos)

Módulo led RGB:



https://wiki.keyestudio.com/Ks0032_keyestudio_RGB_LED_Module



Programa de ejemplo:

niciolizor	
Bucle	
Led RGB	G 10 🔹 Pin B 11 🔹 Color 📒
Esperar 1000 milisegundos	
Led RGB Anodo Común Pin R 9 V Pin	G 10 🔹 Pin B 11 🔹 Color
Esperar 1000 milisegundos	
Led RGB Anodo Común Pin R 9 Pin	G 10 🔹 Pin B 11 🔹 Color
Esperar 1000 milisegundos	
Led RGB Anodo Común Pin R 9 Pin	G 10 🔹 Pin B 11 🔹 Color
Esperar 1000 milisegundos	
Led RGB Anodo T común Pin R 9 T Pin	G 10 🔹 Pin B 11 🔹 Color
Esperar 1000 milisegundos	
Led RGB	G 10 🔹 Pin B 11 🔹 Color
Esperar 1000 milisegundos	
Led RGB	G 10 v Pin B 11 v Color
Esperar 1000 milisegundos	

<u>Relé</u>

El relé es un interruptor controlado electrónicamente. Permite controlar cargas de potencia.

Módulo relé



https://wiki.keyestudio.com/Ks0011 keyestudio 5V Relay Module

Esquema de conexiones:



Programa de ejemplo:



LDR (consola serie)

La fotocélula o LDR (Light Dependent Resistor) es una resistencia que varía según la intensidad de la luz, por lo que nos permite medir el nivel de luz ambiente fácilmente.

Módulo LDR:



https://wiki.keyestudio.com/Ks0028 keyestudio Photocell Sensor



Programa de ejemplo:

Inici	ializar 🔹																			
	🗾 Iniciar E	Baudio	os 960	0																
											÷	+								
E	>_ Enviar (661	Nivel d	le luz (senso	or L DE	2)- 22		Sa	ilto	de	líne	ea							
		1			(ooniot		·/·													
Due																				
- DUG																				
ſ	Establecer n	ivel d	e luz 🔻) = (Nive	l de lu	ız (LD	R)		Ś	F	Pin	A1	¥		0.	.10)23	Y	
F	Establecer n	ivel d	e luz 🔻) = (Nive	el de lu	ız (LD	R)		\$	P	Pin	A1	¥		0.	.10)23	Y	-
E	Establecer n	ivel d	e luz 🔻) = (Nive	el de lu	ız (LD	R)	•	\$	P	Pin	A1	¥		0.	.10)23	V	
E	Establecer n	iivel d	e luz 🔹) = () le luz:	Nive	el de lu	ız (LD Ito de	R) línea	•	*	. F	Pin +	A1	•		0.	.10)23		
E	Establecer n	ivel d	e luz 🔻) = () le luz:	Nive	el de lu	ız (LD Ito de	R) línea			. F 	Pin	A1	•		0.	.10)23		
	Establecer n >- Enviar (>- Enviar (iivel d	e luz 🔹 Nivel d el de lu) = (le luz:	Nive "	el de lu Sa Ilto de	ız (LD Ito de Iínea	R) línea	•		- F 	Pin	A1	•		0.	.10)23		
	Establecer n ≻– Enviar (≻– Enviar (iivel d	e luz ▼ Nivel d el de lu) = (le luz:	Nive " ✓ Sa	el de lu Sa ilto de	ız (LD Ito de Iínea	R) línea	a *		+ + + +	Pin	A1	•	· · · · · · · · · · · · · · · · · · ·	0.	.10)23		
	Establecer n >- Enviar (>- Enviar (Esperar () 2	iivel d	e luz × Nivel d el de lu milise) = () le luz: z v) egundo	Nive " ✓ Sa	el de lu Sa Ito de	Iz (LD Ito de Iínea	R) línea	a 		• • • •	Pin	A1	•	•	0.	.10)23		
	Establecer n >- Enviar (>- Enviar (Esperar () 2	iivel d	e luz 🔹 Nivel d el de lu milise) = (le luz: z •] egundo	Nive " ✓ Sa	el de lu Sa Ito de	Iz (LD Ito de Iínea	R) línea			P 	• • • • • •	A1	•	· · · · · · · · · · · · · · · · · · ·	0.	.10)23		

Consola serie:



Nivel de luz: 65.00 Nivel de luz: 262.00

ArduinoBlocks :: Consola serie

Baudrate: 9600	~	Conectar	Desconectar	Limpiar
			✓ Enviar	
Nivel de luz (sens	or LD	R):		
Nivel de luz: 782.0 Nivel de luz: 781.0	00 00			

LDR (serial plotter)

Módulo LDR:



https://wiki.keyestudio.com/Ks0028_keyestudio_Photocell_Sensor

Esquema de conexiones:



Programa de ejemplo:

In	icializar																								
			_		_																				
	>_ Ini	ciar Bau	idios	9600																					
B	ucle • •	* * * *	+ +	+ + -		-	+		÷	+	÷ ÷	÷	÷	÷	÷	• •	÷	÷	+	+	+	÷	÷	-	
				_													_		_		_			_	
	Estable	cer nive	l de lu	z 🔹 🛛	= 🔘	Ni	ivel	de	luz	: (L	DR)			F	Pin	F	\1	٠		0.	.10)23	•	
	Estable	cer nive	l de lu	z 🔪	= (Ni	ivel	de	luz	: (L	DR)	e	Ø	5	Pin	A	1	•		0.	.10)23	•	
	Estable	cer nive	l de lu	z 🔪	= ()	Ni	ivel	de	luz	: (L	DR)	¢			Pin		41	•		0.	.10)23	•	
	Estable	cer nive Plotter	l de lu	z 🔹	= ()] "	Ni	ivel ⁄alo	de or 🌘	luz	: (L ivel	DR de	luz		¢		Pin		¥1	•		0	.10)23	•	
	Estable	Plotter	l de lu	z 🔻	= ()] "	Ni V	ivel /alo	de or 🌔	luz	: (L ivel	DR de	luz		9		Pin		¥1	•		0	.10)23		
	Establed	cer nive Plotter	l de lu (ˈ ᢩ ٬٬ mili	z 🔻 LUZ segu	= ()] " ndos		ivel ⁄alo	de r 🌘	luz	: (L ivel	DR	luz				Pin		¥1	•	· · · · · · · · · · · · · · · · · · ·	0	.10)23		
	Establed	Plotter	l de lu (ˈ mili	z v LUZ segu	= ()] " ndos		ivel /alo	de or 🌘	luz	: (L ivel	de	luz				Pin	-	¥1	•	•	0	.10)23		

Serial plotter:



ArduinoBlocks :: Serial plotter + Datalogger



х



Х

(si usamos varias series de datos, se descargará un .csv por cada una)

LDR + Relé (encendido automático por nivel de luz)

Módulo LDR:



Módulo Led:





Programa de ejemplo:

nicializar																							
ucle																							
Estable	ecer Ini	vel de		× .		N	ivel	de	luz (ם ו	R)	18	-	Pin	A	1 🔻		0	10	72	3	• I	
LStabit		verue							(,		•								•		
+ si	C' (nive	el de	luz		<	• (3	00					•		• • •	6		•				-
+ si		nive	el de	luz	v Pin	<	· (3 Sets	00 00					•			6		- - -		•	•	
+ si hacer	Led	nive	el de	luz	v Pin	<	▼ (▼ [3 Esta	00) ndo (ON				• • • • •		· · ·	73		-		•	•	
+ si hacer	Led	nive	el de	luz	▼ Pin	<	▼ (▼ [3 Esta	00) ado (ON	 			• • • •	· · · · · · · · · · · · · · · · · · ·	· · ·			-		*	•	
+ si hacer	Led	nive	el de	luz	▼ Pin	5	▼ (▼ [Esta	00 1do (ON	1 •			• • • •	* * * * * *	 			· · · · · · · · · · · · · · · · · · ·		*	• • • •	
+ si hacer sino	Led	nive	el de	luz	y Pin Pin	5	▼ (▼ (▼ (Esta Esta	100 11do (11do (ON				* * * * * * *		· · ·	<u></u>	* * *	• • •			• • • • • •	
+ si hacer sino	Led		el de	luz	Pin	5		Esta Esta	100 11do (11do (ON	↓ ▼ F ▼			* * * * * * * *			2	* * * * * *	• • • • • • • • • •		*		
+ si hacer sino	Led		el de	luz	▼ Pin Pin	5		Esta Esta	300 ado (ado (ON				* * * * * * * * *	+ + + + + + + + + + + + + + + + + + + +			* * * * * * *	•	· · · ·	* * * * * * * * * * * * * * * * * * * *	• • • • • • • •	
+ si hacer sino	Led		el de	luz	v Pin Pin	5		3 Esta Esta	00 ido (ON	 			* * * * * * * * *	* * * * * * *		23	· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •			* * * * * * * *	• • • • • • • • • • •

Control de servo (posicionamiento básico)

El servomotor es un motor de corriente continua con un controlador interno y un sistema de posicionamiento que nos permite situarlo en la posición deseada con una señal PWM externa. Los servomotores normales no tienen rotación continua.

Módulo servo:



https://wiki.keyestudio.com/Ks0194_keyestudio_Micro_Servo

El microservo utilizado permite situarlo entre 0º y 180º







Programa de ejemplo:



Control de servo (movimiento suave)

Módulo servo:



Esquema de conexiones:



Programa de ejemplo 1:

Inicializar																	
Bucle	• • •	· · ·	+ + + +	• •	• •												
Estable	ecer pos	sicion vec	• = es	C (
hacer	Servo	Ì	Pin	9 •	Gra	ados	s	pos	sicio	n 🔻	F	etai	do ((ms)		0	
	Establ	ecer (posic	ion 🔻	2) =	G (p	osici	ion		+ •		1		•	•	
	Espera	ar 📭	50	milis	egur	ndos	;										

Programa de ejemplo 2:

-1-1-11																									
nicializar	1																								
-																									
Bucle																									
Bucle					÷.	ľ.								_						-	ŕ				
contar o	con	i	•	de	sd	e			ha	asta	a (1	80		d	e a	a (1	1	4 4				
contar c	con	i	•	de	sd	e			ha	asta	a (1	80		d	e a	a (1	1		+	•	•	+ + +	+
contar o	con	ĺ	•	de:	sd	e (ha	asta	a (80		d	e a		1	1] (n	* *			0	•
contar o	con Se	i	v V 0	de:	sd	e P	in (9	ha	asta Gra	a (ado	1 s (80 (i		d	e a R	a (Ret	ar	1 do) (n	ns) (0	
contar o	con Se	i	•	de:	sd	e (in (9 ,	ha	asta Gra	a (1 s(80		d	e a R	a (Ret	ar	1 do) (n	ns) (0	
contar o	con Se Es		v o erai	de:	sd }	e P 50	in (9 nilis	ha D (sec	asta Gra	a (ado	1 s (80		d	e a R	a (Ret	ar	1 do	(n	ns) (0	
contar o	con Se Es	i erv spe	v o erai	de:	sd	e P 50	in (9 nili:	ha D (seg	asta Gra gun	a (ado ado:	1 s (80		d	e a	a (Ret	ar	1 do	(n	ns) (0	
contar o	con Se Es	erv spe	v o erai	de:	sd F	e P 50	in (9 nilis	ha D (seg	asta Gra gun	a (ado	s (80		d	e a	a (Ret	ar	1 do	(n	ns) (0	

Programa de ejemplo 3:

icializar																							
ucle																							
contar o	con	i	v	de	sde	e ()	0		ha	asta	ď	1	80		le	a (1	1	н 1			
hacer				>					-	-		6	-		1.				_	*	. (*
nacon	S	erv	0	6		PI	n s	9 ,		Gra	dos		1 I	*]	ŀŀ	et	ar	do	ः (n	ns) (0
				\sim																			
	E	spe	erar		5	0	n	nilis	seç	gun	dos		*		•	+	•	•	- 	•	•		•
	E	spe 1	erai		5	0] m	nilis	seg	jun	dos		+	 	+	•	- - -	•	- 	+ + +	- - -		•
contar o	E	spe 1	erai	de:	5 sde	0) m	nilis 80	seç	guno ha	dos sta			· · ·	de l	+ + +			- 	+ + + + + +	• • •	•	• • •
contar	E	spe 1 i	erai	- 👔	5 sde	0) m	nilis 80	seç	guno ha	dos sta		0		le le	+ + +		1		+ + + + +	* * * *	-	+ + + + +
contar o	E con	spe 1 i erv	erai	de:	5 sde	0 e () Pii) m	nilis 80 9		guno ha Gra	dos sta dos	c C	0		J Je	a (ar	1 do	(n	+ + + +	··· ·· ·· ·· ··		0
contar o hacer	Con S	spe 1 i erv	erai	de:	5 sde	0 Pii) m	nilis 80 9 v		gund ha Gra	dos sta dos				le F	a (Ret	ar	1 do) (n	ns	* * * *		0
contar o	Con S	spe 1 i erv	erai V o	de:	5 sde	0 Pii	m 1 1	nilis 80 9 v	seç seç	gun ha Gra gun	dos sta dos dos		0		de] F	a(Ret	ar	1 do	(n	* * * * *	· · · · · · · · · · · · · · · · · · ·		• • • • •

Control de servo (osciladores)

Módulo servo:





Programa de ejemplo 1:

1 servo, oscilando con una amplitud de +/- 90 ° (180 en total), en un período de 1000 ms (en 1s hace la oscilación completa):

	~			a	_	
Servo-Oscilador	S.	Pin 9 🔹	Período (m	s) 🔹 🖣	100	0
Servo-Oscilador		Pin 9 🔻	Amplitud •	90		
Servo-Oscilador	*	Pin 9 🔻	Fase •	0	· ·	
			· · · · · · ·	· · ·		
ucle						
Servo-Oscilador	à.	Pin 9 🔻	Actualizar	3		

Programa de ejemplo 2 (añadir servo en el pin 10):

2 servos, oscilando +/- 90° en 2000ms y uno desfasado 90° respecto al otro:

Serv	o-Oscilador	*	Pin 9 🔹	Período (ms) 🔹	2000
Serv	o-Oscilador	.	Pin 9 🔻	Amplitud •	90
Serv	o-Oscilador	X	Pin 9 🔻	Fase V	
Serv	o-Oscilador	.	Pin 10 🔹	Período (ms)	2000
Serv	o-Oscilador	à.,	Pin 10 🔻	Amplitud 🔹 📢	90
Serv	o-Oscilador	X	Pin 10 🔻	Fase 🔹 🗘 90	
Bucle					
	o o siledar	~		Astuslinov	
Serv	o-Oscilador	<u>چې او </u>		Actualizar	
Serv	o-Oscilador	*	Pin 10 v	Actualizar •	
0010	0 000111401	ST A	10	rotuanzar	
Motor DC (ventilador)

El módulo ventilador incorpora un pequeño motor DC y un driver para controlarlo permitiendo la inversión de giro y el control de velocidad mediante PWM.

Módulo motor-ventilador:



https://wiki.keyestudio.com/Ks0168_keyestudio_L9110_fan_control_module

Esquema de conexiones:



Programa de ejemplo:



Ejemplo de control del módulo de ventilador:



DHT22 (consola / serial plotter)

El sensor DHT22, es un sensor digiral de temperatura y humedad (mejora del DHT11) que permite leer valores de temperatura entre -40 y 125° (saltos de 0.5°) y valores de humedad relativa del aire entre 0 y 100%



https://wiki.keyestudio.com/KS0430_Keyestudio_DHT22_Temperature_and_Humidity_Sensor

Esquema de conexiones:



Programa de ejemplo:



BUS I2C

El bus I2C permite conectar dispositivos o periféricos en forma de BUS (múltiples dispositivos comparten los 2 mismos cables de señales de comunicación)

El bus I2C se compone de dos seañes: SCL y SDA (y VCC y GND para la alimentación de los dispositivos)

http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/



Dentro del bus cada dispositivo debe configurarse en una dirección diferente para evitar colisiones.

Los dispositivos de distinto tipo, por lo general, llevan direcciones diferentes. Si usamos dos dispositivos iguales en el bus (por ejemplo dos pantallas LCD) debemos cambiar la dirección de los dispositivos para que no estén en la misma dirección I2C.

La configuración de la dirección I2C se suele modificar mediante jumpers, microswitchs, o en algún caso hay que soldar y modificar unos contactos en la placa del sensor.



Pantalla LCD 1602 (16 caracteres ancho x 2 líneas):



https://wiki.kevestudio.com/Ks0061 kevestudio 1602 I2C Module

• Default I2C Address: 0x27

Regulación/Ajuste del contraste:



LCD (textos básicos)

La pantalla LCD permite mostrar caracteres alfanuméricos de forma sencilla en una pantalla de 2 filas y 16 columnas. La conexión se realiza mediante bus I2C.



https://wiki.keyestudio.com/Ks0061_keyestudio_1602_I2C_Module

Programa de ejemplo 1:



Programa de ejemplo 2:



Termómetro con LCD y DHT22



Programa:



Símbolos personalizados

La pantalla LCD permite definir hasta 8 símbolos personalizados.

Para definir un símbolo podemos usar el editor que nos generará el "mapa de bits" que representa la imagen del símbolo.

http://www.arduinoblocks.com/web/help/chareditor







Conexiones:



Programa:

Inicializar	
LCD # 1 Definir Simbolo 1 Defi	01110
LCD # 1 * Definir Símbolo 2 * B00000,B01010,B10101,B10001,B10001,B10001,B	01010.
· · · · · · · · · · · · · · · · · · ·	
Bucle	
LCO # 1 V Imprimir Columna 0 V Fila 0 V Símbolo 1 V	
Esperar 2000 milisegundos	
# 1 Limpiar	
# 1 Imprimir. Columna Que Fila Que Símbolo Que	
Esperar 3000 milisegundos	

Algunas ideas:



Sensor CO2: CSS811

El sensor CSS811 es un sensor que puede detectar una amplia gama de compuestos orgánicos volátiles (TVOC) incluyendo el eCO2 (CO2 equivalente).

Rango medición eCO2: 400 ... 29206 ppm (partes por millón)



https://wiki.keyestudio.com/KS0457_keyestudio_CCS811_Carbon_Dioxide_Air_Quality_Sensor

El sensor se conecta mediante bus I2C, podemos conectarlo directamente al bus I2C de la shield.

Pero como posteriormente utilizaremos la pantalla también, podemos conectarlo a los pines que internamente también están conectados al bus I2C:A4=SDA, A5=SCL



Otra opción posible cuando necesitamos conectar varios dispositivos al bus I2C es usar un HUB I2C para "ramificar el bus":

https://shop.innovadidactic.com/es/standard-perifericos/872-keyestudio-hub-i2c.html



Programa para medir los niveles de eCO2 y TVOC:

(el sensor necesita un tiempo de calentamiento/ajuste , por lo que puede tardar unos minutos en dar niveles correctos)

Inicializar
▶ Iniciar Baudios 9600 ▼
Enviar Sensor CCS811: 77 V Salto de línea
Bucle
Establecer co2_ppm • = C Sensor CO2/TVOC (CCS811)
Establecer co2_mgm3 v = C Sensor CO2/TVOC (CCS811) CO2 (mg/m3) v
Establecer tvoc_ppb • = (Sensor CO2/TVOC (CCS811) TVOC (ppb) •
Enviar (+ - crear texto con (" CO2 (ppm): " Salto de línea
Enviar + - crear texto con CO2 (mg/m3): . Salto de línea
Enviar + - crear texto con + TVOC (ppb): >> Salto de línea
Enviar (" " / Salto de línea
Esperar 5000 milisegundos

Medidor CO2 con LCD y sensor CCS811

Conexiones: (ya que tanto la pantalla LCD como el sensor CCS811 usan I2C, para no tener que ramificarlo o usar un HUB, usamos el "truco" de que los pines I2C (SDA/SCL) corresponden internamente a los pines A4=SDA y A5=SCL, por lo que ya lo tenemos ramificado en dos conexiones)



Conexiones:



Programa 1:

Inisializat												
Inicializat												
# 1 x Iniciar 2x16 x I2C ADDR 0x27 * x												
Bucle												
					_		-			_		
Establecer co2_ppm * = 1, Sensor CO2/TVOC (C	CS8	11)	-	٦Ċ			C	02	(p	pn	n) '	
						_						
			3	2	44							
			3		44							
Lco # 1 · Limpiar			=		<u>4.</u> 4							
Lco # 1 Limpiar	 		3		N.S.			-				
Lco # 1 · Limpiar			1		₩e.			1	-	-		
LCO # 1 Limpiar	" Ni	vel		2 (p	opn	1):	"	1	-	-		
Lco # 1 · Limpiar Lco # 1 · Imprimir Columna 0 · Fila 0 · Columna	" (Ni	vel	= CO:	2 (p	opn	1):	"			-		
LCO # 1 · Limpiar LCO # 1 · Imprimir Columna 0 · Fila 0 · C	" (Ni	vel	1	2 (p	opn	1):	"			-		
Lco # 1 · Limpiar Lco # 1 · Imprimir Columna 0 · Fila 0 · C Lco # 1 · Imprimir Columna 0 · Fila 1 · C	" (Ni ::02_	vel	ی دo:	2 (p	opn	1):	"			-		
Lco # 1 · Limpiar Lco # 1 · Imprimir Columna 0 · Fila 0 · C Lco # 1 · Imprimir Columna 0 · Fila 1 · C	" (Ni co2_	vel	CO:	2 (p	opn	1):	"			· • • •		
Lco # 1 · Limpiar Lco # 1 · Imprimir Columna 0 · Fila 0 · C Lco # 1 · Imprimir Columna 0 · Fila 1 · C	" (Ni co2_	vel ppn		2 (r	pn	1):	"					
Lco # 1 · Limpiar Lco # 1 · Imprimir Columna 0 · Fila 0 · C Lco # 1 · Imprimir Columna 0 · Fila 1 · C Lco # 1 · Imprimir Columna 0 · Fila 1 · C Esperar 5000 milisegundos	" (Ni co2_	vel ppn	= co:	2 (p	pn	1):	"					

Programa 2:

Inicializar
LCD # 1 Iniciar 2x16 I2C ADDR 0x27 * V
Establecer co2_ppm • = 1. Sensor CO2/TVOC (CCS811)
LCD # 1 Limpiar
1 • Imprimir Columna 0 • Fila 0 • (4 Nivel CO2 (ppm); *
LCD # 1 Imprimir Columna 0 Fila 1 Co2_ppm
+ si (co2_ppm •) < • (400)
hacer # 1 V Imprimir Columna 8 V Fila 1 V ()
sino si – C (co2_ppm v) < v (700
hacer # 1 Imprimir Columna 8 Fila 1 I i i bien "
sino si – C (co2_ppm • < • C 2500
hacer # 1 Imprimir Columna 8 Fila 1 Imprimir Columna 8 Fila 1 Imprimir Columna 8 Imprimir
sino # 1 Imprimir Columna 8 Fila 1 1 4 mai ?
Esperar 5000 milisegundos

Semáforo CO2 con Led RGB y sensor CSS811

A la práctica anterior le añadimos un led RGB , para visualmente mostrar el estado de la calidad del aire en función del nivel de CO2

Conexiones:



Progama:

Inicializar
1 • Iniciar 2x16 • I2C ADDR 0x27 * •
Led RGB
Bucle
Establecer co2_ppm • = 1 Sensor CO2/TVOC (CCS811)
1 Limpiar
LCD # 1 V Imprimir Columna 0 V Fila 0 V (Nivel CO2 (ppm): "
+ si (<u>co2_ppm)</u> < (400)
hacer # 1 Imprimir Columna 8 Fila 1 Imprimir Columna 8 Imprimir
Led RGB
sino si – C (co2_ppm) <) [700]
hacer # 1 Imprimir Columna 8 Fila 1 V bien ?
Led RGB
sino si – C (co2_ppm) < C (2500)
hacer # 1 • Imprimir Columna 8 • Fila 1 • • regular *
Led RGB Ánodo Común Pin R 9 Pin G 10 Pin B 11 Color
Led RGB
Esperar 5000 milisegundos

Mando a distancia IR

Receptor IR



https://wiki.keyestudio.com/Ks0026_keyestudio_Digital_IR_Receiver_Module

https://wiki.keyestudio.com/Ks0088 New Infrared IR Wireless Remote Control Module Kits for Ar duino

El receptor es capaz de recibir y decodificar señales de luz infrarroja, procedente de mandos a distancia.

Podemos obtener los códigos de los botones de los mandos de casi cualquier marca (protocolos: NEC, Sony, RC5, RC6, Panasonic, JVC, ...)

El mando que viene incluido con el sensor es un mando genérico con protocolo RC5:



Conexiones:



Programa para "capturar" códigos de mandos (prueba con el mando de la TV, aire acondicionado, ...) y mostrarlos por la consola:

Iniciali	zar																									
	Inicia	or Poudio	0000																							
<u> </u>			5 9000																							
	÷																									
Bucle																										
Est	tablece	r codigo	recibido	v =	C	Re	ece	pto	r de	IR	(Te	xto	HE	EX)		200		۵.	-	6	ŝ	Ρ	Pin	1	1 🔻	
																30	~		V		<u></u>					
+	si	Cod	igo recib	ido	2	dif	ere	nte	de	• (1	' () 2	•				+		*	*	•	•		· .	-
+ had	si cer	Cod Envia	igo recib r (] " (ido Códi	go (dif dete	ere	nte ado	de :) '	v (? (' (Sal) ?: Ito d) le l	îne	a	•	* * * *	* * *	* * * *	+ + +	•	•	+ · + · + ·	· ·	•
+ had	si cer	Cod - Envia	igo recib r ((r (ido Códi ligo r	go (ecit	dif dete	ecta	ente ado	de :)	▼ (2 (alto	de	• (Sal Iín	to d	le I	íne	* a		* * * * * * *	* * * * *	* * * * * * *	* * * * * * * * * * * *	• • • • •	* * *	* · * · * ·	· · ·	•

Para cada tecla que queramos usar de un mando a distancia debemos apuntar los códigos recibidos:

(Asegúrate de pulsar varias veces y ver el código que normalmente se repite y descarta errores. Los códigos tiop FFFFFF son para indicar que la tecla no se ha soltado)

ArduinoBlocks :: Consola serie
Baudrate: 9600 V Conectar Desconectar Limpiar
► Enviar
Código detectado: 00FF6897 Código detectado: 00FF6897 Código detectado: FFFFFFF Código detectado: 000000FF Código detectado: FFFFFFF Código detectado: 00FF6897 Código detectado: FFFFFFF Código detectado: 00FF6897 Código detectado: 00FF6897 Código detectado: FFFFFFF Código detectado: FFFFFFF Código detectado: FFFFFFF
Fecla 1: 00FF6897
Fecla 2: 00FF9867
Fecla 3: 00FFB04F
Fecla flecha arriba: 00FF629D
Fecla OK: 00FF02FD

<u>Piano IR</u>

Cada tecla del mando a distancia hará sonar una nota musical diferente en el zumbador y pondrá el led RGB de un color distinto.

Conexiones:



Progama:



<u>Servo IR</u>

Control de la posición del servo con el mando IR.

Tecla 1: posición 0°

- Tecla 2: posición 90°
- Tecla 3: posición 180°
- Tecla flecha izquierda: disminuye 10°

Tecla flecha derecha: aumenta 10°

Conexiones:



Programa:



Sensor humedad suelo - Riego automático



https://wiki.keyestudio.com/Ks0049 keyestudio Soil Humidity Sensor

Utilizaremos el sensor para detectar un nivel bajo de humedad y activar una bomba de riego automáticamente (activación de la bomba mediante un relé).

Ejemplo de bomba de agua:





Conexiones:



Ejemplo de esquema completo:



Programa:

icializar							
icle 👘							
E-t-bla				D:		0.4000	
Estable	cer numeda suelo 🔰 = 🚺 Sonda de r	iumedad		, Pin	A2 V	01023	5
			399 C				
+ si	humeda suelo 🔪 < 🔨 🕻 150						
hacer	Relé 🧆 Pin 12 🗸 Estado	ON V					
14001							
	and the second sec						
	Esperar 💭 10000 milisegundos						
	Relé 🧀 Pin 12 🗸 Estado	OFF 🔻					
	· · · · · · · · · · · · · · · · · · ·						
	Esperar 600000 milisegundos						
~							
Espera	5000 milisegundos						

Sensor magnético (Velocímetro bicicleta)

Sensor de campo magnético (efecto Hall):



https://wiki.keyestudio.com/Ks0020_keyestudio_Hall_Magnetic_Sensor

El sensor se actica o desactiva al acercar un campo magnético (imán).



Necesitamos saber cuando se activa o desactiva (flanco ascendente o descendente de la activación), para eso usaremos un bloque que genera un evento externo (interrupción) cuando se detecta un cambio en el sensor (además es capaz de detectar cambios a velocidades muy altas)

Flanco de subida o de bajada al cambiar el estado de un sensor:



Para detectar el evento del flanco de subida (de 0 a 5v, al acercar un imán al senor):

In	ter	ru	рс	iói	n	Pir	۱ (2	•	RI	SI	NG	3	7	
(\sim					1								2
	\rightarrow	\sim	\sim		\sim			+		\sim	+			+	

Siguiendo este sistema, vamos a realizar un velocímetro de bicicleta. Cada vez que el sensor genera un flanco de subida quiere decir que la rueda ha dado una vuelta completa, sabiendo el tiempo que ha tardado desde la última detección y la longitud de la circunferencia de la rueda podemos calcular la velocidad:





Cálculo de la velocidad (en Km/h):

velocidad = (longitud_rueda_metros / 1000) / (tiempo_transcurrido_ms / 1000 / 3600) Simplificando:

velocidad = longitud_rueda_metros / (tiempo_transcurrido_ms / 3600)

Para visualizar la información se usará la pantalla LCD.

Conexiones:



Programa:



Sensor de sonido

Permite, mediante un micrófono, detectar el nivel de sonido ambiente.



https://wiki.keyestudio.com/KS0035_Microphone_Sound_Sensor_with_Potentiometer

Filtro de la señal de sonido (filtro mediana):

S -> sensor de sonido puro

M -> señal filtrada

ArduinoBlocks :: Serial plotter + Datalogger



×

Semáforo de sonido:

Inicializar
Led RGB
Filtro mediana # 1 Iniciar para [10] muestras
Bucle
Ejecutar cada (100) ms
Establecer nivel sonido = C Nivel de sonido Pin A0 • 01023 •
Establecer sonido filtrado = Filtro mediana # T Filtrar inivel sonido •
Ejecutar cada (2000) ms
+ si (Sonido filtrado - < 10
hacer Led RGB Ánodo v común Pin R 9 v Pin G 10 v Pin B 11 v Color
sino si – 🗘 🖌 sonido filtrado 🔹 🤇 🔽
hacer Led RGB Ánodo v común Pin R 9 v Pin G 10 v Pin B 11 v Color
sino Led RGB Ánodo común Pin R 9 Pin G 10 Pin B 11 Color

Funciones

A la hora de realizar programas grandes es aconsejable dividir en tareas el programa. Podemos usar funciones para crear bloques de código que realice alguna función en concreto, y así "llamarlo" o "ejecutarlo" de una forma más clara y sencilla.



Ejemplo 1:







Ejemplo 2:



Tareas / Multitarea

Arduino no implementa un sistema multitarea, sólo ejecuta una cosa a la vez, sin embargo mediante técnicas software podemos simular el ejecutar varias cosas a la vez, o por lo menos que compartan el tiempo y parezca que lo hacen a la vez.

1) Método "casero".

La idea es desglosar todo en tareas, cada tarea debe ser rápida y NO BLOQUEAR la ejecución (PROHIBIDO LOS BLOQUES "esperar" y los bucles extra largos o que se queden esperando una condición)



2) Método usando la librería "multitask freeRTOS" un poco limitada en Arduino pero que podemos hacer funcionar. <u>https://drive.google.com/file/d/1r-oo8KUpNBySFMyEHYkjaltZV9eg0NPV/view</u>



SmartHome: conexionado + programación básica

https://wiki.keyestudio.com/KS0085 Keyestudio Smart Home Kit for Arduino#Assembled Guide

Conexiones completas:



Funcionalidades a implementar en el programa para hacer la casa autónoma:

-Mensaje inicial de bienvenida en LCD

-Control con mando IR:

Tecla arriba -> ventilador ON

Tecla abajo -> ventilador OFF

Teclas 1,2,3 -> melodías RTTTL

- -Pulsador 1 -> abrir/cerrar ventana
- -Pulsador 2 -> abrir/cerrar puerta

-El sensor de movimiento encenderá automáticamente el led blanco

-Si el sensor de humedad de suelo detecta poca humedad y no llueve , se activa el relé para activar el riego

-Si el nivel de luz es bajo, se enciende automáticamente la luz amarilla

-Cada 5s se muestra por la pantalla LCD el nivel de CO2 , la temperatura y la humedad










IMPORTANTE: Poner los servos en posición 90º antes de montarlos:



Programa:



+ para	ovimiento	
+ si	C Detector de movimiento (PIR) Pin 2	
hacer	Led Pin 13 V Estado ON V	
sino	Led Pin 13 Stado OFF	





+ para iniciar	
Establecer puerta 1 abierta y = (falso y	· · · · · · · · · · · ·
Servo 🕉 Pin 🦭 Grados 🖬 Ángulo 90°	Retardo (ms)
Establecer puerta 2 abierta = (falso -	· · · · · · · · · · · ·
Servo 🥉 Pin 10 🗸 Grados 🕻 Ángulo 🕐	Retardo (ms)
Establecer mov detectado V = C falso V	

+ para	puertas	
+ si	C Pu	ilsador 🥵 Pin 4 💌 se ha pulsado 🔹 Invertir
hacer	+ si	puerta 1 abierta 🕇
	hacer	Servo 🕉 Pin 🦭 Grados 🕌 Ángulo 90° Retardo (ms) 🚺 👔
		Establecer puerta 1 abierta y = (falso y
	sino	Servo 🕉 Pin 🦭 Grados 🗐 Ángulo 180° Retardo (ms) 🚺 🚺
		Establecer puerta 1 abierta = (verdadero V
+ si	(Pu	ilsador 🥵 Pin 8 🔹 se ha pulsado 🔹 Invertir
hacer	+ si	puerta 2 abierta
	hacer	Servo 🕉 Pin 10 Grados Ángulo 0° Retardo (ms) 10
		Establecer puerta 2 abierta V = C falso V
	sino	Servo 🕉 Pin 10 🗸 Grados 🗍 Ángulo 180° Retardo (ms) 🚺 🛛
		Establecer puerta 2 abierta = (verdadero v

+ para LCD saludo																			
LCD # 1 Limpiar																			
LCD # 1 Imprimir	Cc	olui	nn	a (0	•	Fila	a (C) 🔻	6	/lal	et.	.In	no	v.	4.(),	,	



iniciar LCD # 1 Iniciar 2x16 I2C ADDR 0x27 * • iniciar LCD saludo Esperar 5000 milisegundos cle mando IR puertas movimiento riego Ejecutar cada 1000 ms luz ldr Ejecutar cada 5000 ms		1.1																
# 1 Iniciar 2x16 I2C ADDR 0x27 * Inicial Iniciar LCD saludo Esperar 5000 milisegundos cle mando IR puertas movimiento riego Ejecutar cada 1000 ms LCD info	licializar						1			1				1				
iniciar LCD saludo Esperar 5000 milisegundos cle mando IR puertas movimiento riego Ejecutar cada 1000 ms luz Idr Ejecutar cada 5000 ms	LCD	# 1 🔹	Inic	ciar	2	x1	6	Y	12	2C	A	DD	R	0	x2	7	*	7
iniciar LCD saludo Esperar (5000) milisegundos cle mando IR puertas movimiento riego Ejecutar cada (1000) ms luz Idr Ejecutar cada (5000) ms																		
LCD saludo Esperar (5000 milisegundos cle mando IR puertas movimiento riego Ejecutar cada (1000 ms luz Idr Ejecutar cada (5000 ms	iniciar	1.1.1				1	i.	1			1	÷.	1	ï	1	ĵ,	1	1
Esperar 5000 milisegundos cle mando IR puertas movimiento riego Ejecutar cada 1000 ms luz Idr Ejecutar cada 5000 ms																		
Esperar 5000 milisegundos cle mando IR puertas movimiento riego Ejecutar cada 1000 ms luz Idr Ejecutar cada 5000 ms	LCD sa	aludo																
cle mando IR puertas movimiento riego Ejecutar cada (1000 ms luz ldr Ejecutar cada (5000 ms	Espera	r / 50	000	n	nilio	-		nd	100									
cle mando IR puertas movimiento riego Ejecutar cada (1000) ms luz ldr Ejecutar cada (5000) ms	сэрсга		100			SC!	yu		103	5								
cle mando IR puertas movimiento riego Ejecutar cada (1000) ms luz Idr Ejecutar cada (15000) ms																		
cle mando IR puertas movimiento riego Ejecutar cada (1000) ms luz ldr Ejecutar cada (5000) ms																		
e mando IR puertas movimiento riego Ejecutar cada (1000 ms luz Idr Ejecutar cada (5000 ms																		
mando IR puertas movimiento riego Ejecutar cada 1000 ms luz ldr Ejecutar cada 5000 ms	cle																	
mando IR puertas movimiento riego Ejecutar cada (1000) ms luz Idr Ejecutar cada (5000) ms																		
uertas novimiento ego jecutar cada (1000) ms luz ldr jecutar cada (5000) ms LCD info	iando	IR																
movimiento riego Ejecutar cada (1000) ms luz ldr Ejecutar cada (5000) ms LCD info	nuerta																	
novimiento iego Ejecutar cada 1000 ms Iuz Idr Ejecutar cada 5000 ms																		
riego Ejecutar cada (1000) ms luz ldr Ejecutar cada (5000) ms LCD info	movimi	ento																
Ejecutar cada 1000 ms luz ldr Ejecutar cada 5000 ms LCD info	riego																	
Ejecutar cada (1000 ms luz ldr Ejecutar cada (5000 ms LCD info	- age								Ċ.									
luz ldr Ejecutar cada (5000 ms	Ejecuta	ar cada	IC P	100	00		ms											
LCD info				-	-													
Ejecutar cada 5000 ms	luz I	dr																
Ejecutar cada (5000 ms									1									
LCD info	Fiecuta	ar cada		500	00	1	ms											
LCD info			L		_													
	LCE) info																

Comunicaciones - Serie / Bluetooth

Arduino permite enviar o recibir información a través de su conexión serie (pines 0,1)

Se utiliza un protocolo UART:

https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html





Originalmente Arduino tenía un puerto serie (tipo RS232) para conectar al PC, hoy en día ese puerto está prácticamente extinguido en los ordenadores y en su lugar se utiliza el USB que sería la evolución de este tipo de puerto (aunque mucho más avanzado, pero al fin y al cabo una comunicación de datos en serie de la misma forma). Arduino incorpora un chip que convierte del standard UART TTL a USB (FTDI, CH340G, CP2102,...)





1) Comunicación serie con cable

La comunicación serie tiene dos propósitos:

-Permitir el envío del programa para que el propio Arduino lo reciba y se autoprograme la memoria interna de programa (se hace con el "Uploader" dentro del ArduinoBlocks-Connector)

-Intercambiar información entre Arduino y el PC (consola o aplicación específica, en el caso de la consola de ArduinoBlocks el intercambio de datos se realiza con la ayuda del Connector)

A la hora de intercambiar información entre Arduino y el PC,Móvil,... (cualquier dispositivo capaz de conectarse por comunicación serie con la placa), podemos trabajar a nivel de bytes o de texto (el texto es una secuencia de bytes de caracteres imprimibles)

Por simplicidad y por ser más "*entendible*" vamos a ver varios ejemplos de intercambio de datos Arduino <-> PC/Móvil a nivel de texto y cómo interpretar esos datos.

Inicia	alizar										
	Liniciar	Baudios	9600 🔻								
E	Establecer	valor de p	orueba 🔻] = (3.14	16					
E	Establecer	texto de p	orueba 🔻) = ("[Hola	que t	al '	,		
Bucl	le	* * * *					+ +			-	
Bucl	e Envia	· (· · · U r	n texto de	e prue	ba "		Salto	o de	lín	• ea	
Bucl	e Envia - Envia	C " Ur Valor (n texto de de prueb	e prue	ba "	alto d	Salto e líne	o de ea	lín	ea	
Bucl	E Envian ► Envian ► Envian	valor of texto of	n texto de de prueb de prueb	e prue a V	ba " ✔ sa	alto d	Salto e líne e líne	o de ea ea	lín	* * *	
Bucl	le Enviai ≻– Enviai ≻– Enviai ≻– Enviai	Valor of texto of 5000	n texto de de prueb de prueb nilisegune	e prue	ba " ✔ Sa ✔ Sa	alto d	Salto e líne e líne	o de ea ea	lín	- ea 	

1.1) Ejemplo: Enviar datos Arduino -> PC

ArduinoBlocks :: Consola serie

Baudrate: 9600 v	Conectar	Desconectar	Limpiar
		✓ Enviar	
Un texto de prueba			
Hola que tal			

1.2) Recibir datos Arduino <- PC (texto)

iciclizer																				
icializar				+ + +																
	niciar F	Baudios	960	0 -																
~		Juuuloe	000																	
2- EI	nviar (_ " E	spera	ndo d	atos		\checkmark	Sa	alto	de	lín	ea								
le 🐘																				
			Dotoo	rooibi	idoo2															
T SI	5	ک <mark>-</mark>	Dalos	тесірі	uosr															
•											-									
nacer	Esta	blecer	datos	; • =	4		кес	IDIL	tex	to		на	sta	a s	ali	0	de	nea	a	
																- P.				
			F-1 11					••		0	lto	de	s lí							
	► > _	Enviar		Date	os rec	ibido	S:	· ·		56	ano	ue	; II	ne	a	1.1				
		Enviar	<u> </u>	Date	os rec	ibido	s:			56	anto	uc	; 11	ne	a	-				
	>	Enviar		Date	os rec	ibido Solt	S:	lín		56		*	*	ne	а +	4 4				
	>_ >_	Enviar		Date atos	os rec	ibido Salt	s: o de	: lín	lea	56		* *	: II : :	ne	а *					
	2	Enviar		Date atos	os rec	ibido Salt	s: o de	: lín	iea	56		+ + +	; II • •	ne	а + +					

ArduinoBlocks :: Consola serie

hola Cenviar	Baudrate: 9600 v	Conectar	Desconectar	Limpiar
Esperando datos	hola		► Enviar	
	Esperando datos			

ArduinoBlocks :: Consola serie

Baudrate: 9600 V	Conectar	Desconectar	Limpiar
145		- Enviar	
Esperando datos Datos recibidos:hola			

1.2) Recibir datos Arduino <- PC (números)

aializar	
cializar	
S Ir	iciar Baudios 9600 V
Ъ Е	nviar 🕻 🕊 Esperando números 😕 🔽 Salto de línea
	<u></u>
	• • • • • • • • • • • • • • • • • • • •
olo	
cie	
+ si	ြ 🚬 ¿Datos recibidos?
hacer	Establecer dato recibido 🔹 = 🖡 🛌 Recibir como número 🗸 Hasta salto de línea
	Enviar Número reicibido: 🥂 🕑 Salto de línea
	Enviar dato recibido
	Enviar dato recibido 🗸 🗸 Salto de línea
	Enviar dato recibido 🔹 🗸 Salto de línea

ArduinoBlocks :: Consola serie

Baudrate: 9600 V	Conectar	Desconectar	Limpiar
123.45		✓ Enviar	
Esperando números Número reicibido: 45.00	0		
Número reicibido: 9867 Número reicibido: 3450	7.00).45		
Número reicibido: 123.4	45		

El programa anterior se podría de forma similiar:



ArduinoBlocks :: Consola serie

Baudrate: 9600 v	Conectar	Desconectar	Limpiar
965		✓ Emviar	
Esperando números			
Número recibido: 123.0 Número recibido: 450.0	00 65		

Aplicación de ejemplo: "Adivina el número"

http://www.arduinoblocks.com/web/project/2004

2) Comunicación Bluetooth

Los módulos Bluetooth para Arduino permiten simular una conexión serie punto a punto de forma inalámbrica a través de la conexión Bluetooth, de forma que la manera de trabajar es prácticamente igual que con la comunicación serie por cable.

Los módulos Bluetooth más comunes son:

HC-05 / HC-06 (Bluetooth "normal")



HM-10 (Bluetooth BLE 4.0)





El módulo bluetooth se puede usar en pines diferentes al 0,1 para implementar un puerto serie por software y así dejar el puerto serie por cable utilizable también.



Si usamos el módulo Bluetooth conectado a los pines 0,1 para usar el puerto serie integrado de Arduino, debemos configurar de la siguiente forma:



(el pin RX de Arduino se conecta con el TX del módulo Bluetooth, y viceversa)

2.1) Enviar datos Bluetooth -> PC/Móvil

Programa para enviar Temperatura y humedad via Bluetooth:



Visualizamos los datos desde una aplicación de "Consola Bluetooth o Terminal Bluetooth" en un móvil Android:

https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal&hl=es&gl=US



Configuración para enviar y recibir sin saltos de líneas en la aplicación de Terminal Bluetooth:



2.2) Recibir datos Bluetooth <- PC/Móvil

Vamos a implementar un sencillo sistema para enviar comandos simples desde la terminal Bluetooth del móvil y según el comando encendemos o apagamos un led conectado a Arduino.

Comandos:

LED1ON / LED1OFF -> Enciende o apaga el led coenctado en el pin 13

LED2ON / LED2OFF -> Enciende o apaga el led conectado en el pin 5

t si	ک کے کمان کے کا کھن کے کہ
hacer	Establecer comando 🔹 = 🕻 👔 Recibir texto 🛑 Hasta salto de línea
	+ si (comando • igual a • (4 LED1ON ?)
	hacer Led Pin 13 • Estado ON •
	+ si (comando • igual a • (" LED10FF "
	hacer Led Pin 13 T Estado OFF
	+ si (comando • igual a • (" LED2ON "
	hacer Led Pin 5 Estado ON
	+ si (comando • igual a • (" LED2OFF "
	hacer Led Pin 5 Estado OFF

Desde la terminal Bluerooth:



2.3) Protocolo simple de comunicación

En muchas ocasiones necesitamos realizar comandos con algún parámetro, por ejemplo siguiendo el ejemplo anterior podríamos indicar el led y su nivel de intensidad de 0 a 255 (en vez de ON/OFF)

O por ejemplo para situar un servomotor en una posición en concreta entre 0 y 180 grados.

Podemos crear un sistema simple de comando+parámetro con el siguiente formato:

COMANDO/VALOR#

Para que funcione correctamente, siempre los comandos enviados deben seguir ese formato:

Comando "PUERTA" con parámetro entre 0 y 180

Ejemplo: PUERTA/90#

Comando "LED" con parámetro entre 0 a 255

Ejemplo: LED/200#

Programa para recibir y procesar el comando + valor según el formato indicado:

Inicializar	
🚯 👘	
Bucle	
+ si	C 🚯 ¿Datos recibidos?
hacer	Establecer comando • = C 🚯 Recibir texto hasta C Valor ASCII
	Establecer parametro texto 🔹 = 🕻 🚯 Recibir texto hasta 🕻 Valor ASCII # 🔹
	Establecer parametro • = • Texto a número 🕻 parametro texto •
	+ si (comando • igual a • (• PUERTA ?)
	hacer Servo 🕉 Pin 🤊 Grados parametro 🕇 Retardo (ms) 🚺
	+ si (comando 🔻 igual a 🗸 (" LED "
	hacer Led intensidad (PWM) Pin 5 Valor parametro

Ejemplo para poner el led a intensidad 100:

≡	Termi	inal		-0	-	i :
					k	
M 1	M2	МЗ	M4	M5	M6	М7
LED/1	00#					>

Ejemplo para mover la puerta a 120 grados:

≡	Term	inal		-	F	:
M 1	M2	МЗ	M4	М5	M6	M7
PUERT	A/120	#				>

Bluetooth + SmartHome Kit App

Módulo Bluetooth HM-10 (Bluetooth 4.0 BLE)



Programa de ejemplo para funcionar con la APP (versión código Arduino IDE)

https://wiki.keyestudio.com/KS0085_Keyestudio_Smart_Home_Kit_for_Arduino#Project_15.EF.BC.9A Multi-purpose_Smart_Home_Kit



App de ejemplo:

GooglePlay (Android): <u>https://play.google.com/store/apps/details?id=com.keyestudio.iot_keyes</u>

AppStore (iOS): <u>https://apps.apple.com/es/app/iot-keyes/id1487578236</u>

Comandos enviados por la aplicación:

- 'a' -> luz blanca ON
- 'b' -> luz blanca OFF
- 'c' -> relé ON
- 'd' -> relé OFF
- 'e' -> música 1
- 'f' -> música 2

'h' -> nivel de luz (LDR)

- 'i' -> nivel de CO2 (devuelve el valor)
- 'j' -> nivel humedad suelo (devuelve valor)
- 'k' -> nivel de lluvia (devuelve valor)
- 'l' -> ventana abierta (180°)
- 'm' -> ventana cerrada (0°)
- 'n' -> puerta abierta (180°)
- 'o' -> puerta cerrada (0°)
- 'p' -> led amarillo ON
- 'q' -> led amarillo OFF
- 'r' -> ventilador ON
- 's' -> ventilador OFF
- 't' + [valor] -> ventana a posición [valor]
- 'u' + [valor] -> puerta a posición [valor]
- 'v' + [valor] -> led amarillo intensidad PWM = [valor]
- 'w' + [valor] -> motor ventilador velocidad PWM = [valor]

Programa ArduinoBlocks:

http://www.arduinoblocks.com/web/project/710937

Establecer temperatura · = 1 DHT-22 Temperatura · · Pin (A0 · ·
Establecer humedad * = 1 DHT-22 >> Humedad % * Pn (AD *
LLTP # Limplar
tee # 1 • Imprimir Columna 0 • Fils 0 • 4 Femp(C); **
tos # 1 · Imprimir Columna 10 · Fla 0 · Formatear número Vitemperatura · con 1 · decimales
4 1 Imprimir Columna D - Fila 1 - 4 Hun. (%): ??
tte # 1 • Imprimir Columna 10 • Fila 1 • Formatear número # humodad • con 0 • decimales
👻 si 🛛 🕽 sensor_mov 🐑
hseer teo # 1 Imprimir Columna 15 File 1 Simbolo 1

+ para saludo LCD		
Lco # 1 * Limplar		
tto # 1 • Imprimir	Columna 💽 Fila 💽	" Maleta Innov 4.0 "
Lco # 1 mprimir	Columna 💽 Fila 💶 I	" Smart Home IoT "



Pulsador Pin 4 • se ha pulsado • Invertir

 # 1 • Imprimir Columna 0 • File 0 • * ** Auto LDR: **

 # 1 • Imprimir Columna 0 • File 1 • **

Establecor Sensor mox • = ... Datactor de movimiento (PIR)

+ ei euto mov * hacer + si Sensor_mov * hacer Led Pin 13 * Estado ON *

+ si auto Idr • hacer + si

sino Led 🛷 Pin 13 * Estado OFF *

Led Pin 5 • Estado ON •

ecer nivel_ldr v = L Nivel de luz (LDR) 🛷 Pin 🗛 🔹 0..1023 •

Pin 5 · Estado OFF ·

Establecer auto_kir • • • no • auto_kir •

+ si Pulsador Pin () se ha pulsado invertir hacer Establicos fauto mov = 1, no i auto mov +

Esperar 1 2000 milisegundos





Bluetooth + AppInventor

AppInventor es un entorno online de programación visual por bloques que permite crear de forma sencilla aplicaciones móviles compatibles con Android (y ahora también iOS, aunque con algunas limitaciones)

https://appinventor.mit.edu/



AppInventor es compatible con Bleutooth Standarda, para trabajar con BLE debemos instalar un plugin:

https://mit-cml.github.io/extensions/

http://iot.appinventor.mit.edu/assets/resources/edu.mit.appinventor.ble-20200828.aix

1)Ejemplo de una aplicación sencilla para controlar el led encendido o apagado con comandos simples:

AppInventor:

https://drive.google.com/file/d/1ZViHS1HGzNNt_nvboZjLfkMG6g7x5WxH/view?usp=sharing

	Proyectos • Conectar • Generar • Settings • Ayuda • Mis proyectos View Trash G	uia Informar de un problema Espai	iol • arduinoblocks@gmail.com •
MaletaInnova_LED_ONOFF	Screen1 • Anadir ventana Eliminar ventana Publish to Gallery		Diseñador Bloques
Paleta	Visor	Componentes	Propiedades
Best Components Interfaz de usuario Botón Casilla DeVerificación Selector DeFecha Selector DeFecha Eliqueta Selector DeFecha Selector DeLista Selector DeLista Campo DeContraseña Deslizador Desplegable Switch Campo DeTexto Selector DeHora Selector DeHora Selector DeHora 	Mostrar en el Visor los componentes ocultos Tamaño del teléfono (505.320) Maleta Innovación 4.0 Buscar dispositivos BT añadr eleme Conectar Led blanco on oFF	 Screen1 Etiqueta2 brhBuscar Etiqueta1 DisposiciónHorizontal3 brhConcetar DisposiciónHorizontal4 Etiqueta3 Iton To DisposiciónHorizontal5 Etiqueta4 Zon Izonf BluetoothLE1 Cambiar nombre Borrar 	I2on ColorDeFondo ■ Por defecto Habilitado Image Imagen Negren Imagen Negren Forma por defecto •
VisorWeb 🤊		Medios Subir archivo	✓ Tevto
Disposición			ON
Medios	Componentes no visibles		PosiciónDelTexto



Podemos probar la aplicación en el dispositivo mediante la aplicación "Al Companion" (una vez completada y probada podemos generar la aplicación e instalarla)

https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3&hl=es_419&gl=US

		Proyectos •	Conectar • Generar • S	Settings • Ayuda •
laletaInnova_LED_C	DNOFF	Screen1 • A	Al Companion	ublish to Gallery
Bloques		Visor	USB	
Integrados	^		Refresh Companion Screen	
Control			Reiniciar conexión	-
Lógica			Reiniciar completamente	
Matemáticas			ejecutar poner BluetoothLE	. NullTerminateStri
Texto				

Aplicación ejecutándose en Android:

Maleta Innovación 4.0					
Buscar dispo	Buscar dispositivos BT				
Conectado co	orrectar	nente			
E0:62:34:CD:BC:5D HMSoft -54 🔹					
Conectar	Conectar Desconec tar				
Led blanco	ON	OFF			
Led amarillo	ON	OFF			
	Buscar dispo Conectado co 2:34:CD:BC:5E Conectar Led blanco Led amarillo	Buscar dispositivos Conectado correctar 2:34:CD:BC:5D HMS Conectar Desc ta Led blanco ON Led ON amarillo	Buscar dispositivos BT Conectado correctamente 2:34:CD:BC:5D HMSoft -54 Conectar Desconec tar Led blanco ON OFF amarillo		

y programa en Arduino:



2) Ejemplo de una aplicación sencilla para controlar la intensidad de un led com comando+valor:

https://drive.google.com/file/d/1dRRscU6GJ20I29VQ295Q29qYbEWy7ntc/view?usp=sharing

_Mostrar Tamaño d	en el Visor los lel teléfono (5	componentes of 05,320) V	cultos	
	•			
			\$1 B	9:48
	Maleta	Innovación	4.0	
	Buscar	dispositivos I	вт	
		-		
	a	ñadir eleme 🗸		
	Conecta	ar Desco	nectar	
		Led		
		Puerta		
		_		
	\bigtriangledown	0		

cuando	Deslizador1 .PosiciónCambiada		
posició	nDelPulgar		
ejecutar	Ilamar BluetoothLE1 .WriteString	gs	
	serviceUu	uid 🚺	" 0000FFE0-0000-1000-8000-00805F9B34FB "
	characteristicUu	uid 🔰	" 0000FFE1-0000-1000-8000-00805F9B34FB "
	utf1	16 🚺	falso 🔻
	value	es 🔰	😟 unir (" (LED/) "
			(tomar posiciónDelPulgar v
			(" # "
			·

cuando Deslizador2 V. PosiciónCambiada		
posiciónDelPulgar		
ejecutar	Ilamar BluetoothLE1 .WriteStrings	
	serviceUuid 🖡	" 0000FFE0-0000-1000-8000-00805F9B34FB "
	characteristicUuid 🖡	" 0000FFE1-0000-1000-8000-00805F9B34FB "
	utf16 🖡	falso
	values 🖡	😟 unir 🖯 " (PUERTA/) "
		tomar posiciónDelPulgar 🕥
		() " # "



Otras placas y kits

Arduino UNO es perfecto para iniciarse en la robótica educativa y junto al kit smarthome tenemos un punto de partida con posibilidad de multitud de proyectos y funcionalidades gracias a la variedad de sensores, actuadores y periféricos que incorpora así como la conectividad bluetooth.

Disponemos de kits al mismo nivel con distintas aplicaciones, y también totalmente compatibles con ArduinoBlocks:

• **3dBot**: imprimible en 3d, siguelíneas, evita-obstáculos, control bluetooth, etc.



https://shop.innovadidactic.com/es/placas-kits-y-robots/64-kit-imagina-arduino-3dbot-para-arduinobloc ks-y-snap4arduino.html

-Manual de prácticas 3dBot con ArduinoBlocks:

https://drive.google.com/file/d/1ARwRJfFNYqoybMLQeNTnBPpu5zYU0sto/view?usp=sharing

• **KeyBot**: similar al 3dBot pero con soporte metálico (no es imprimible en 3d, y no lleva Arduino reutilizable, es una placa personalizada que incluye el microcontrolador Arduino y conexiones tipo RJ11 para ampliar)



https://shop.innovadidactic.com/es/easy-plug-placas-shields-y-kits/851-keyestudio-kit-robot-keybot-pis ta.html

-Manual KeyBot con ArduinoBlocks:

https://drive.google.com/file/d/1u78lleYch46GRNQqxNfEb-kh3ZPeiQYa/view?usp=sharing

• Kit Arduino + TDR Steam: incuye un Arduino UNO y una shield educativa con múltiples sensores, una pantalla LCD, un mando IR, sensor de sonido externo, y posibilidad de ampliación



https://shop.innovadidactic.com/es/placas-kits-y-robots/1445-kit-imagina-tdr-steam-basado-en-arduino .html

-Manual TDR-STEAM con ArduinoBlocks:

https://drive.google.com/file/d/1d0E5d3q5bRT2IFGZSeYi0npw4KE6XMtY/view?usp=sharing

• ESP32 STEAMakers: Una placa muy potencia y con grandes posibilidades para el loT (internet de las cosas). Incorpora un microcontrolador ESP32 de doble núcleo, WiFi, Bluetooth clásico y BLE. Formato compatible Arduino y con conexiones modulares.



https://shop.innovadidactic.com/es/standard-placas-shields-y-kits/1567-placa-esp32-steamakers.html

-Prácticas ESP32 STEAMakers con ArduinoBlocks:

https://drive.google.com/drive/u/1/folders/1RRSVxQpGqOphfk_Koh1Gvfbjt1JMZo1b

-Videos demostrativos:

https://www.youtube.com/watch?v=MQjIEI7I4ik&list=PL1pKD-Bz2QBAgfy580m8OaQ2Z60v6DOhC