

Instituto Tecnológico de Zitácuaro

Tema 1:

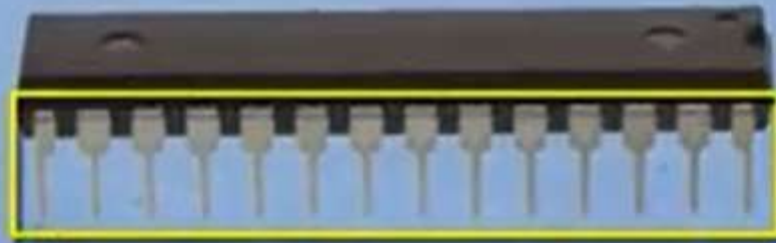
Arquitectura básica de un microcontrolador

Asesor

Dr. Eduardo López Sandoval

Puertos

Pueden ser usados para recibir (entrada) o entregar (salida) información en forma de voltaje.



Puertos

Pueden ser usados para recibir (entrada) o entregar (salida) información en forma de voltaje.



Voltaje

Puertos

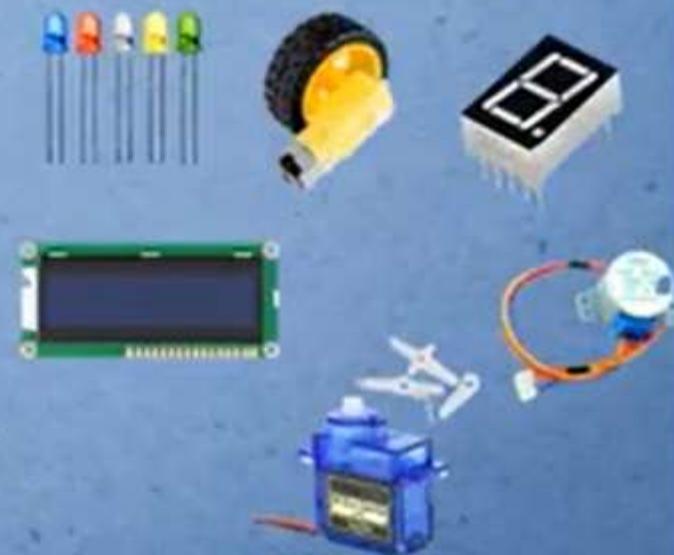
Pueden ser usados para recibir (entrada) o entregar (salida) información en forma de voltaje.



Sensores



Interactúan con el exterior



Actuadores

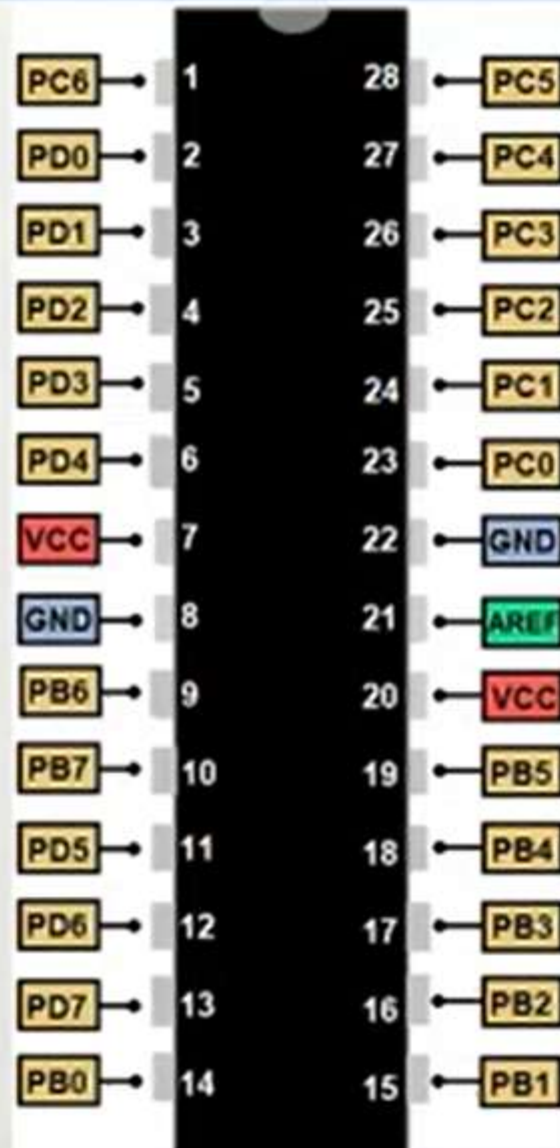
Periféricos del ATmega328p



PORTB: 8 bits

PORTD: 8 bits

PORTC: 7 bits



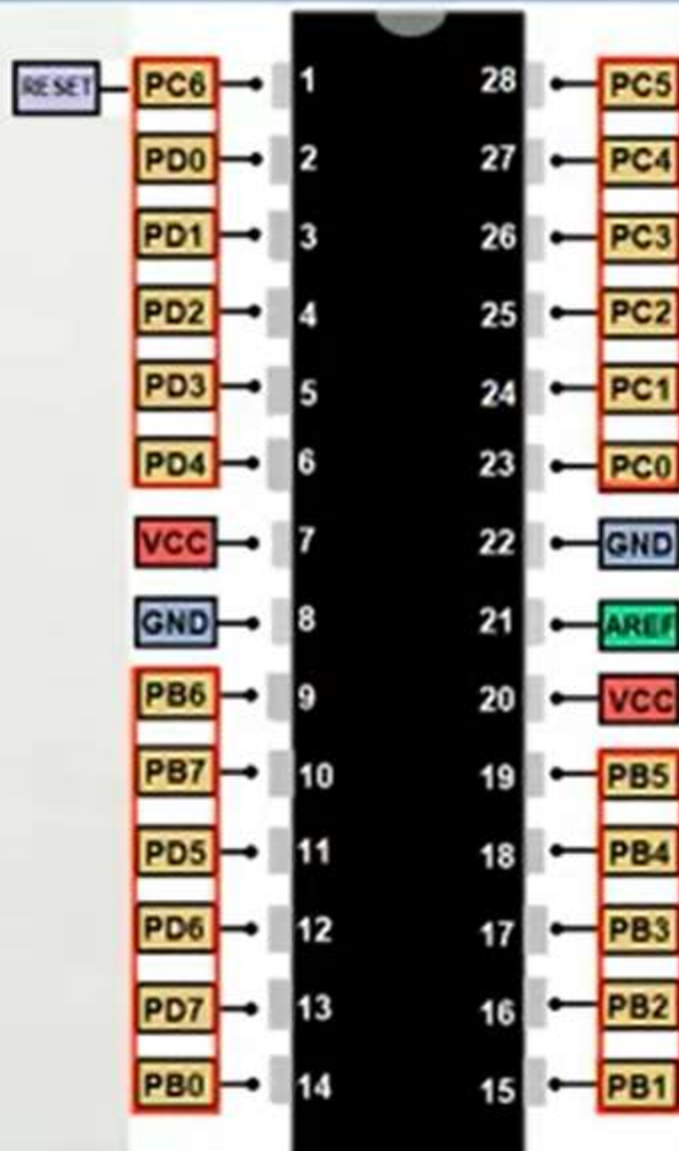
Periféricos del ATmega328p



PORTB: 8 bits

PORTD: 8 bits

PORTC: 7 bits



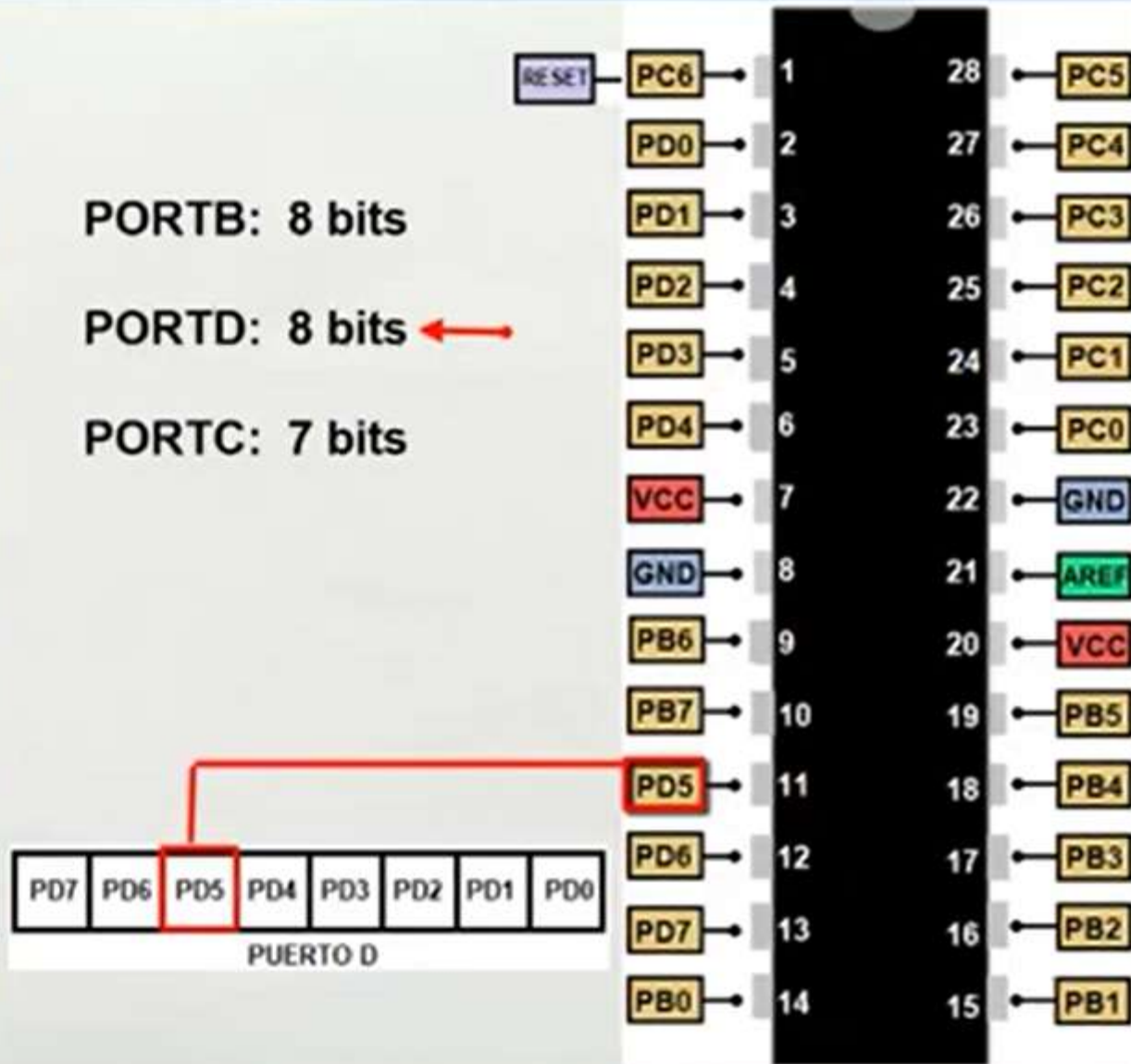
Periféricos del ATmega328p



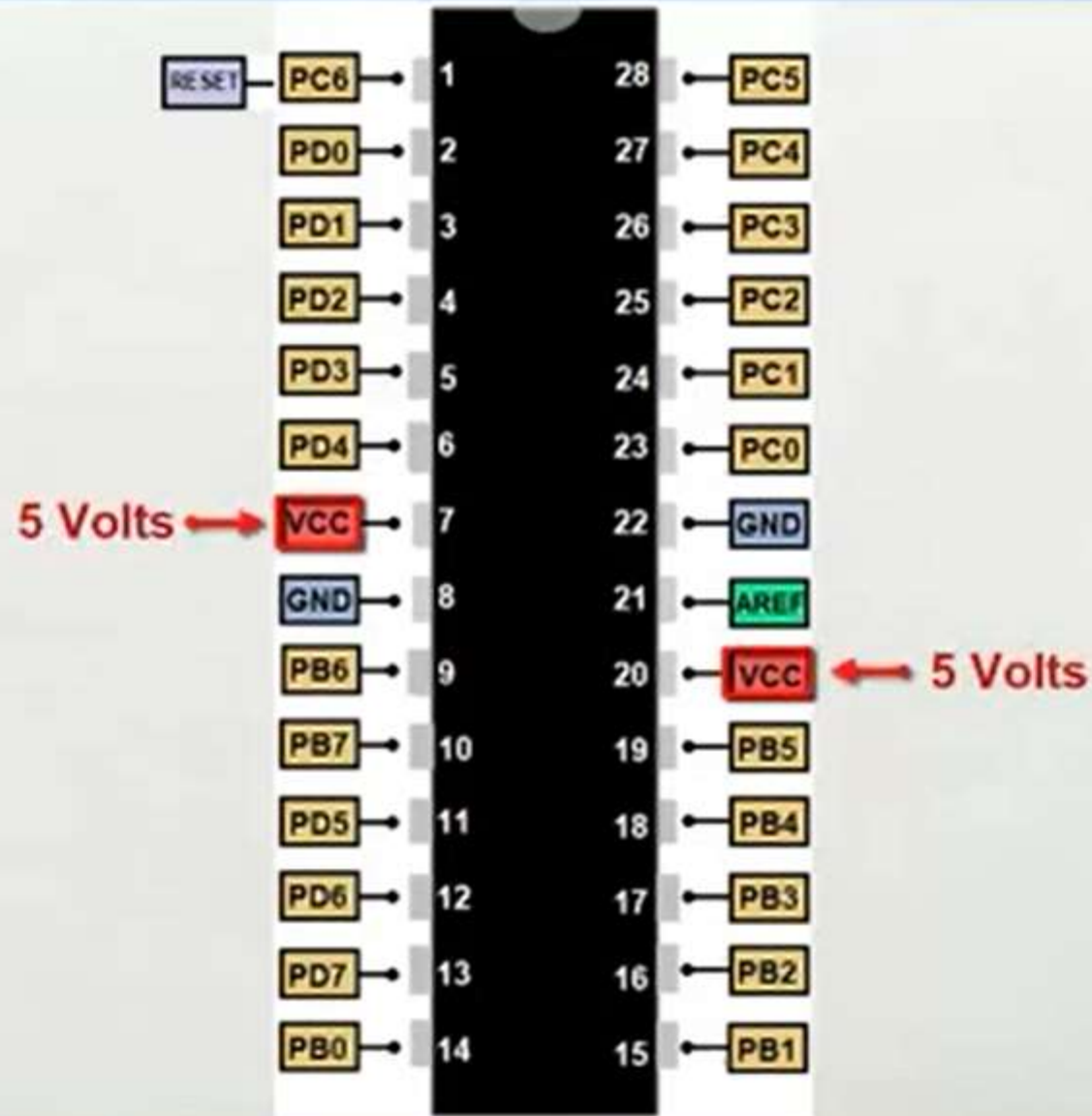
PORTB: 8 bits

PORTD: 8 bits ←

PORTC: 7 bits



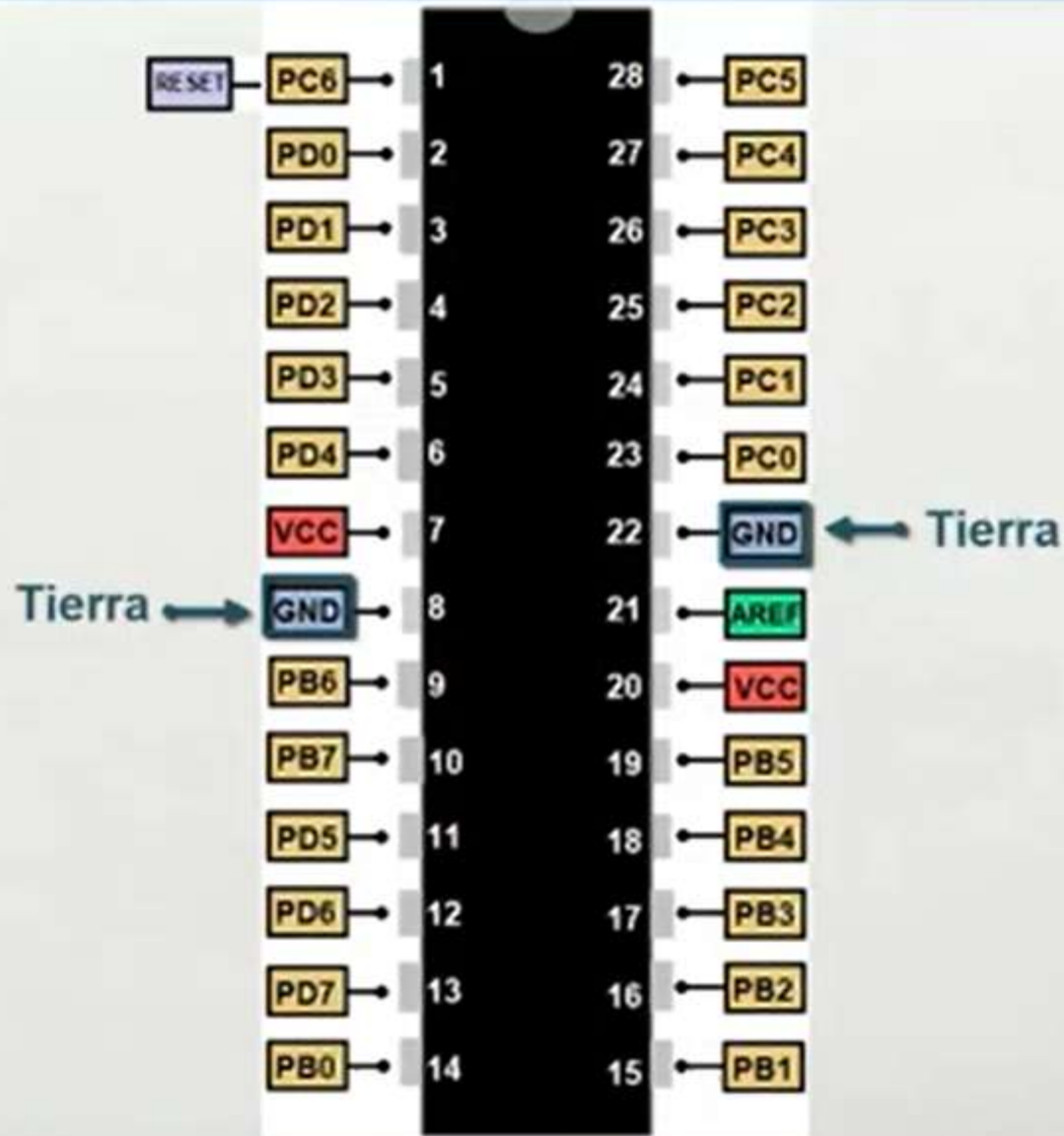
Periféricos del ATmega328p



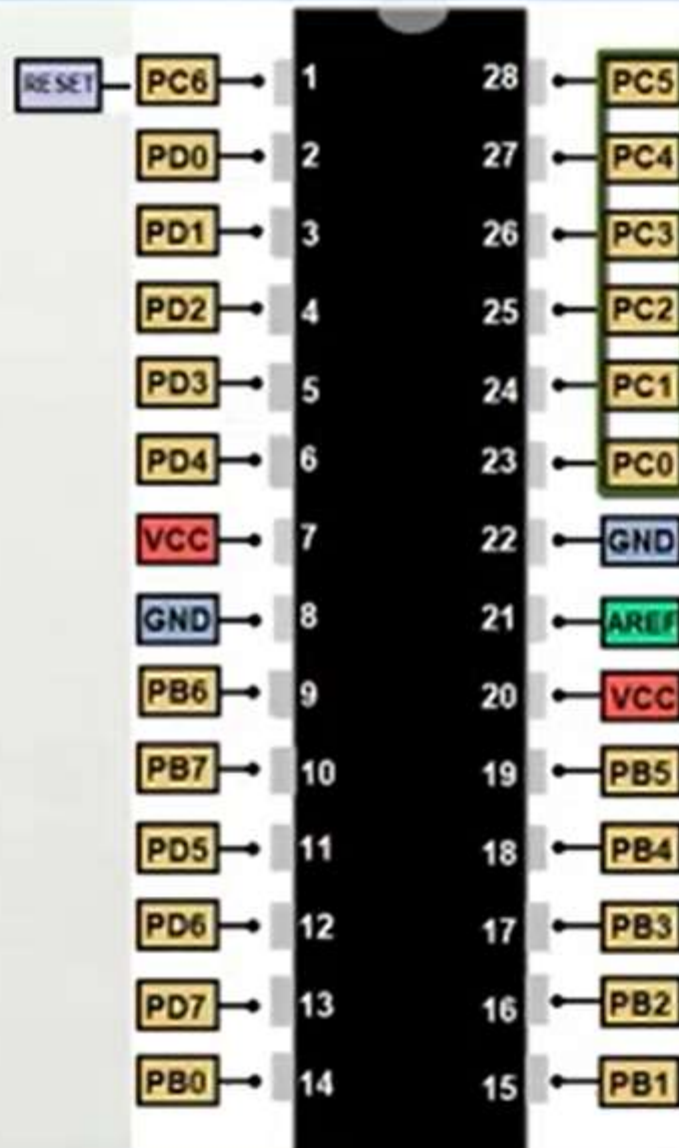
Periféricos del ATmega328p



Periféricos del ATmega328p



Periféricos del ATmega328p

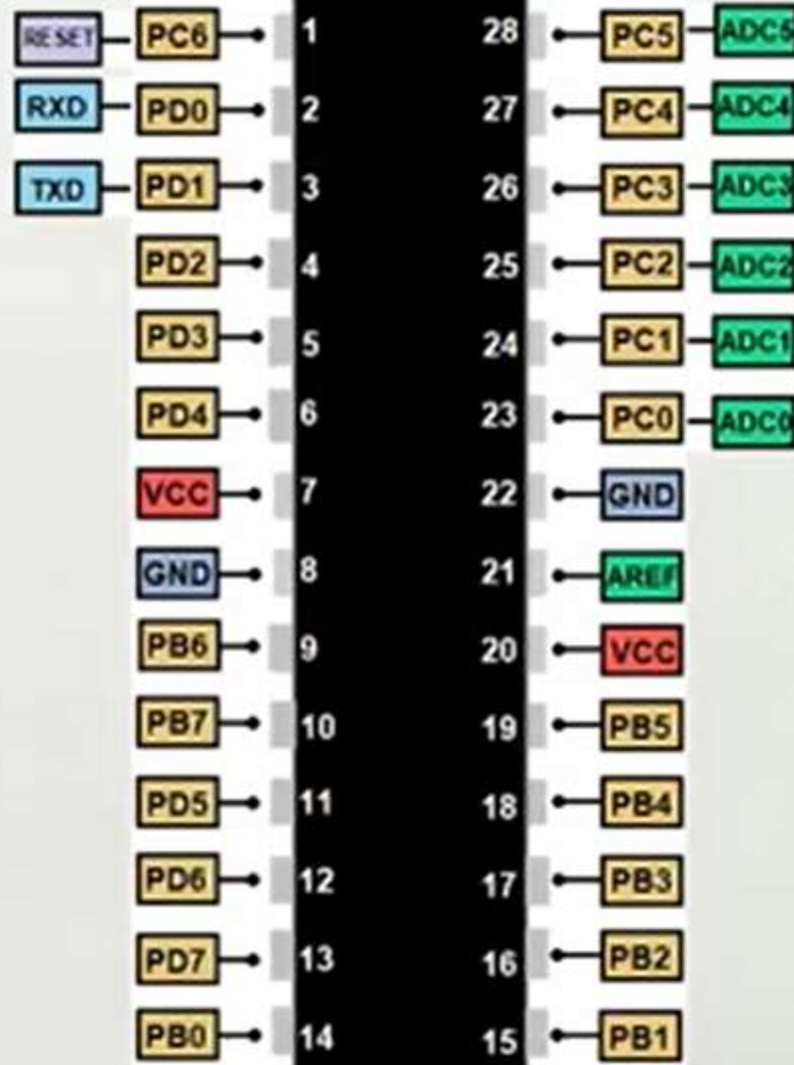


Convertidores
Analógico-Digital

Periféricos del ATmega328p



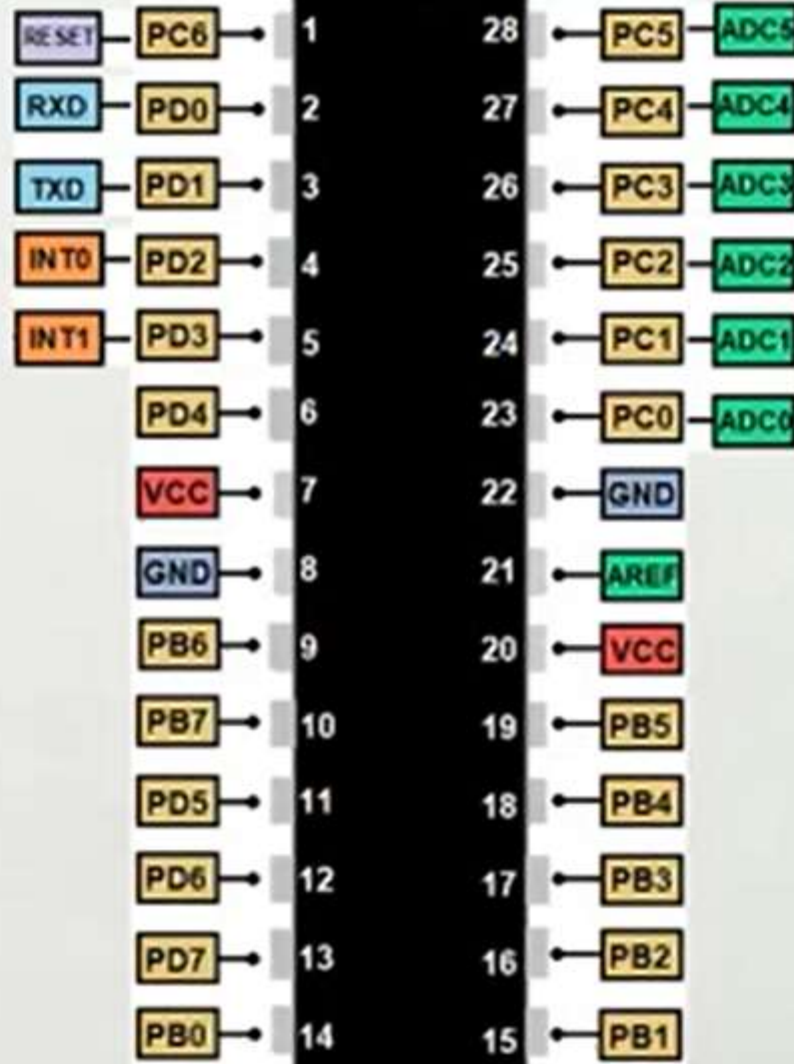
USART



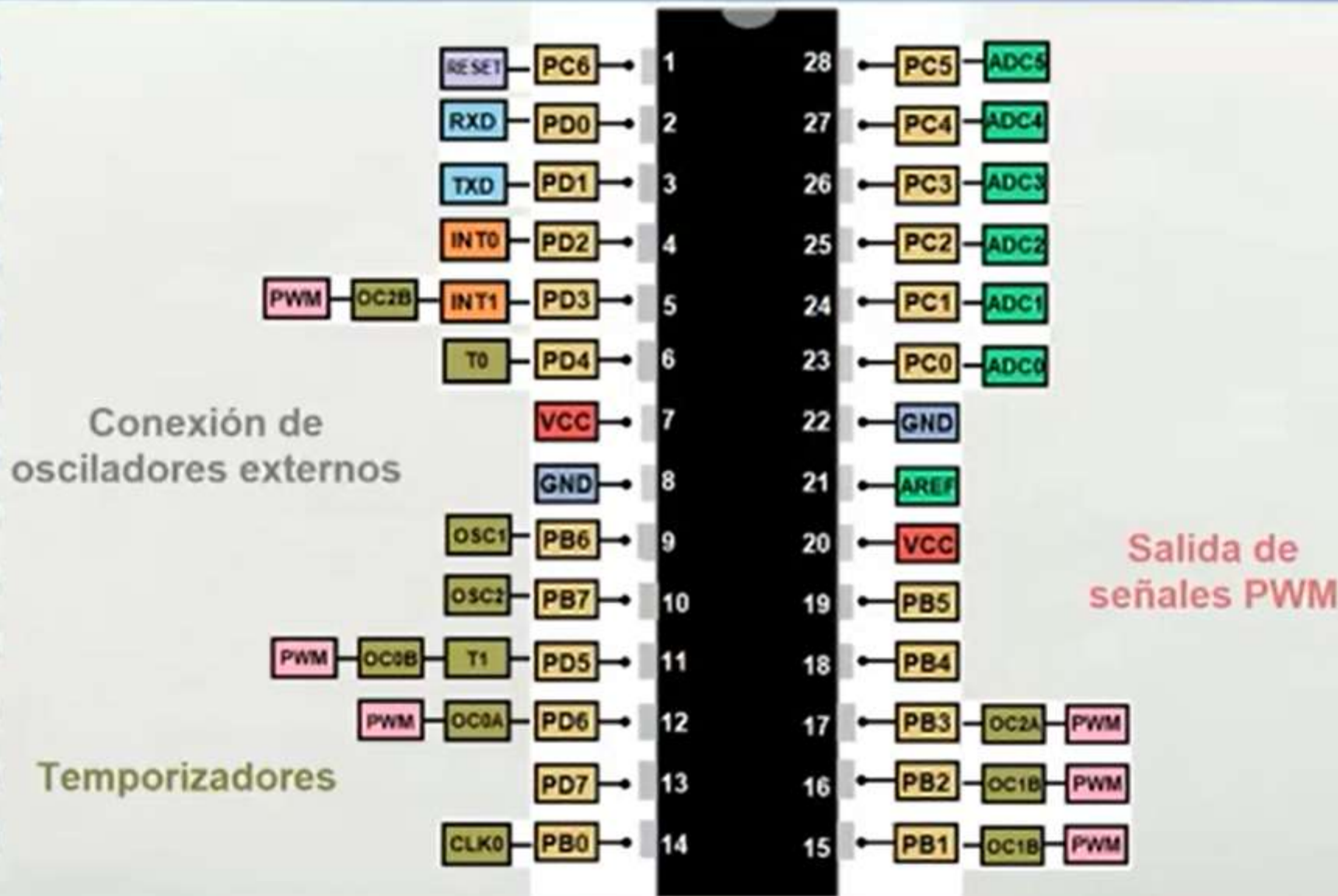
Periféricos del ATmega328p



Interrupciones
externas



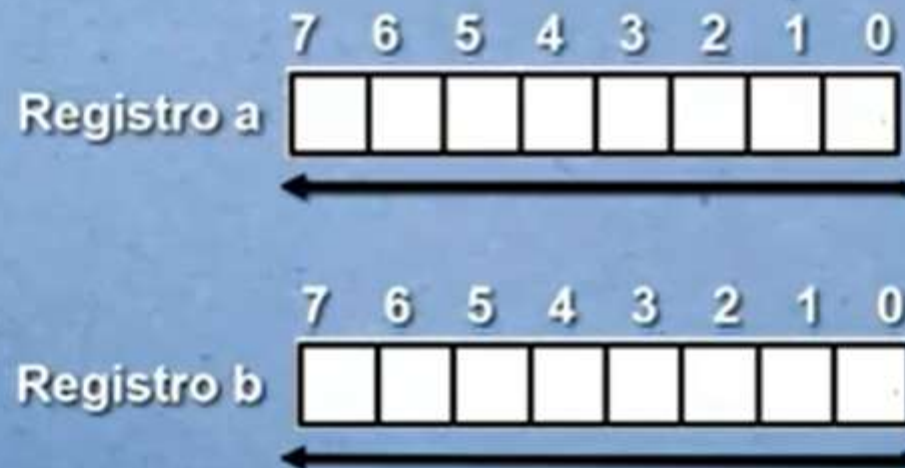
Periféricos del ATmega328p



Registros



Espacios físicos dentro del microprocesador usados para guardar información que tienen un propósito específico.



Tipos de registros



- Register space



R1
R2
R3
R4
⋮
R15
R16
R17
⋮
R31

IN/OUT		Load/Store
0x0000 – 0x001F	32 registers	0x0000 – 0x001F
	64 I/O registers	0x0020 – 0x005F
	160 Ext I/O registers	0x0060 – 0x00FF
	Internal SRAM (2048x8)	0x0100 0x08FF

Tipos de registros



- Register space
- I/O Memory



IN/OUT		Load/Store
	32 registers	0x0000 – 0x001F
0x0000 – 0x001F	64 I/O registers	0x0020 – 0x005F
	160 Ext I/O registers	0x0060 – 0x00FF
	Internal SRAM (2048x8)	0x0100
		0x08FF

Tipos de registros

- Register space
- Extended I/O Memory
- I/O Memory



IN/OUT		Load/Store
	32 registers	0x0000 – 0x001F
0x0000 – 0x001F	64 I/O registers	0x0020 – 0x005F
	160 Ext I/O registers	0x0060 – 0x00FF
	Internal SRAM (2048x8)	0x0100
		0x08FF

Tipos de registros

- Register space
- Extended I/O Memory
- I/O Memory



IN/OUT	Load/Store
0x0000 – 0x001F	32 registers 0x0000 – 0x001F
	64 I/O registers 0x0020 – 0x005F
	160 Ext I/O registers 0x0060 – 0x00FF
Internal SRAM (2048x8)	0x0100
	0x08FF

Tipos de registros

- Register space
- Extended I/O Memory
- I/O Memory



IN/OUT		Load/Store
	32 registers	0x0000 – 0x001F
0x0000 – 0x001F	64 I/O registers	0x0020 – 0x005F
	160 Ext I/O registers	0x0060 – 0x00FF
	Internal SRAM (2048x8)	0x0100
		0x08FF

Configuración de entradas y salidas

DDRx			Función
0			Entrada
1			Salida

Configuración de entradas y salidas

DDRx	PORTx		Función
0	0		Entrada pull-up desactivado.
	1		Entrada pull-up activado.
1			

Configuración de entradas y salidas

DDRx	PORTx		Función
0			
1	Su valor es enviado a los pines del puerto.		Salida

Configuración de entradas y salidas

DDRx	PORTx	PINx	Función
0	0	Contiene los niveles lógicos aplicados a los pines del puerto.	Entrada pull-up desactivado.
	1		Entrada pull-up activado.
1	Su valor es enviado a los pines del puerto.		Salida

Ejemplo:

Configuración como
Entrada al puerto D

DDRD

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

PORTD

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

PIND

0	0	1	0	0	1	0	0
---	---	---	---	---	---	---	---

Entrada
sin pull-up



Entrada
con pull-up



Salida



1

1

Ejemplo:

Configuración como
Salida al puerto B

DDRB

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

PORTB

1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---

Entrada
sin pull-up



Entrada
con pull-up



Salida



↓
1

↓
1

↓
1

↓
1

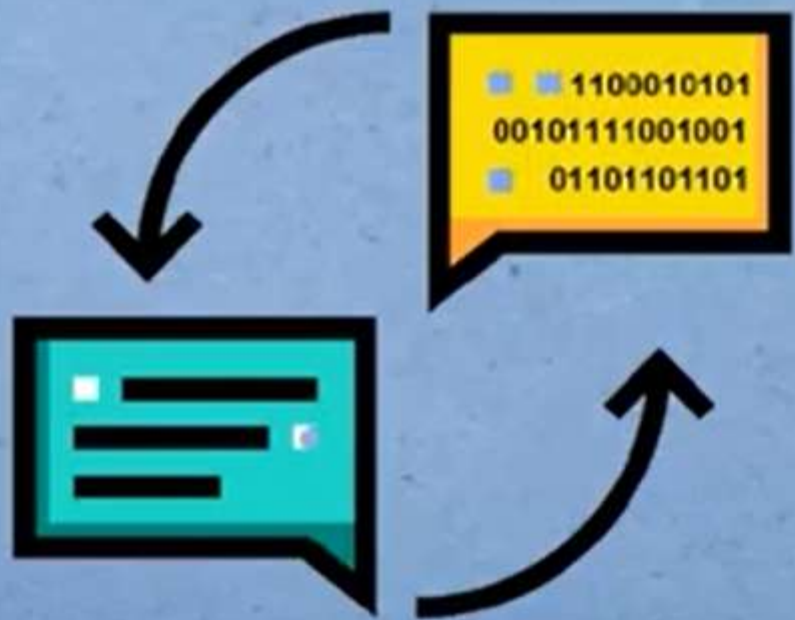
Código máquina



Microcontrolador

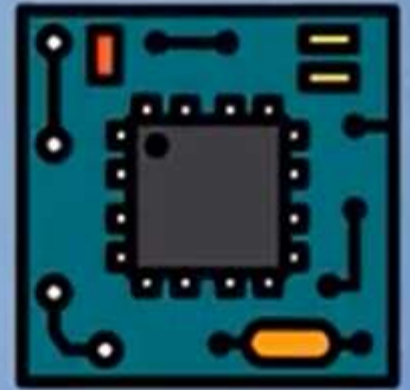
Ensamblador

Lenguaje de bajo nivel, expresa las instrucciones de una forma más natural al hombre a la vez que muy cercana al microcontrolador.



```
1100010101  
00101111001001  
01101101101
```

```
1100010101  
00101111001001  
01101101101
```



Mnemónicos

Caracteres alfanuméricos que simbolizan las órdenes o tareas a realizar.



Código fuente.



Escrito por el programador.

Código máquina.



Se carga al microcontrolador.



Conjunto de instrucciones (AVR - 8 bits)



- Instrucciones aritméticas y lógicas.
- Instrucciones de desvío.
- Instrucciones de transferencia de datos.
- Instrucciones de bit y prueba de bit.



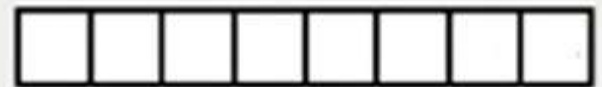
Nomenclatura de instrucciones



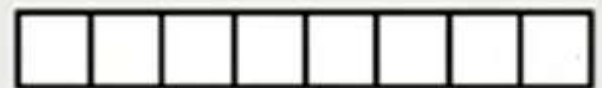
Registros y operandos	
Rd	Registro destino del bloque de registros de trabajo.
Rr	Registro fuente del bloque de registros de trabajo.
R	Resultado después de que una instrucción es ejecutada.
K	Dato inmediato (constante).
k	Dirección de memoria.
b	Bit de un registro I/O (7-0).
s	Bit en el registro de estado (SREG).
X,Y,Z	Registros de direccionamiento indirecto o de palabra
P	Registros de bloque I/O.
q	Desplazamiento (6-bit) por direccionamiento directo.



DDRx



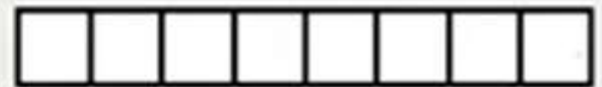
R16



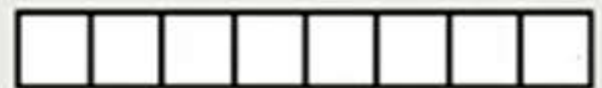
Valor 1



PORTx



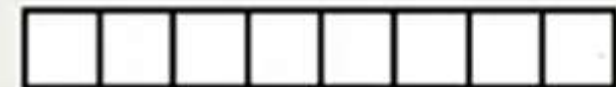
R17



Valor 2



PINx



R18



Valor 3



DDRx

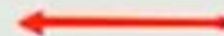
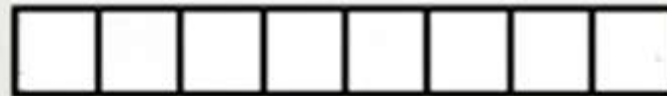
PORTx

PINx



Mnemónico	Operandos	Descripción
LDI	Rd, K	Carga de un dato inmediato.

[R16 - R31]



Valor de 8 bits

DDRx

PORTx

PINx



Mnemónico	Operandos	Descripción
OUT	P Rr	Cargar en I/O un registro de trabajo

Registro I/O

Mnemónico	Operandos	Descripción
STS	(k) Rr	Almacena directamente en SRAM

Dirección de memoria

DDRx

PORTx

PINx



Memory I/O

OUT

Memory I/O extended

STS

IN/OUT		Load/Store
0x0000 – 0x001F	32 registers	0x0000 – 0x001F
	64 I/O registers	0x0020 – 0x005F
	160 Ext I/O registers	0x0060 – 0x00FF
	Internal SRAM (2048x8)	0x0100 0x08FF

DDRx

PORTx

PINx



Memory I/O

"Periféricos"

Memory I/O extended

"Periféricos
en memoria
extendida"

IN/OUT

Load/Store

0x0000 – 0x001F

32 registers

0x0000 – 0x001F

64 I/O registers

0x0020 – 0x005F

160 Ext I/O registers

0x0060 – 0x00FF

Internal SRAM
(2048x8)

0x0100

0x08FF

DDRx

PORTx

PINx



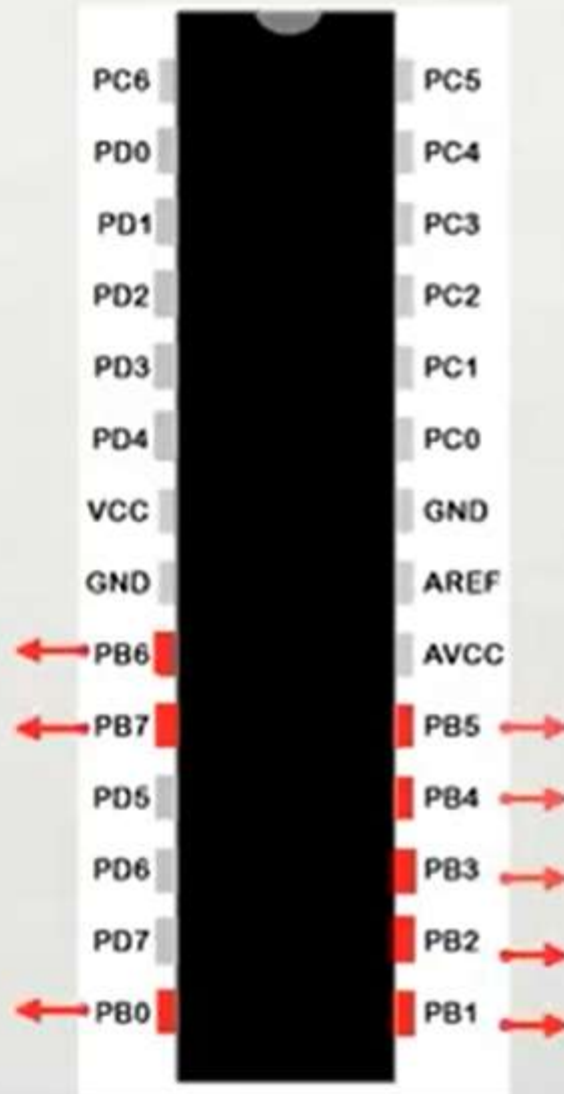
REGISTRO	DIR
PINB	0x03
DDRB	0x04
PORTB	0x05
PINC	0x06
DDRC	0x07
PORTC	0x08
PIND	0x09
DDRD	0x0a
PORTD	0x0b



Memory I/O

Ejemplo:

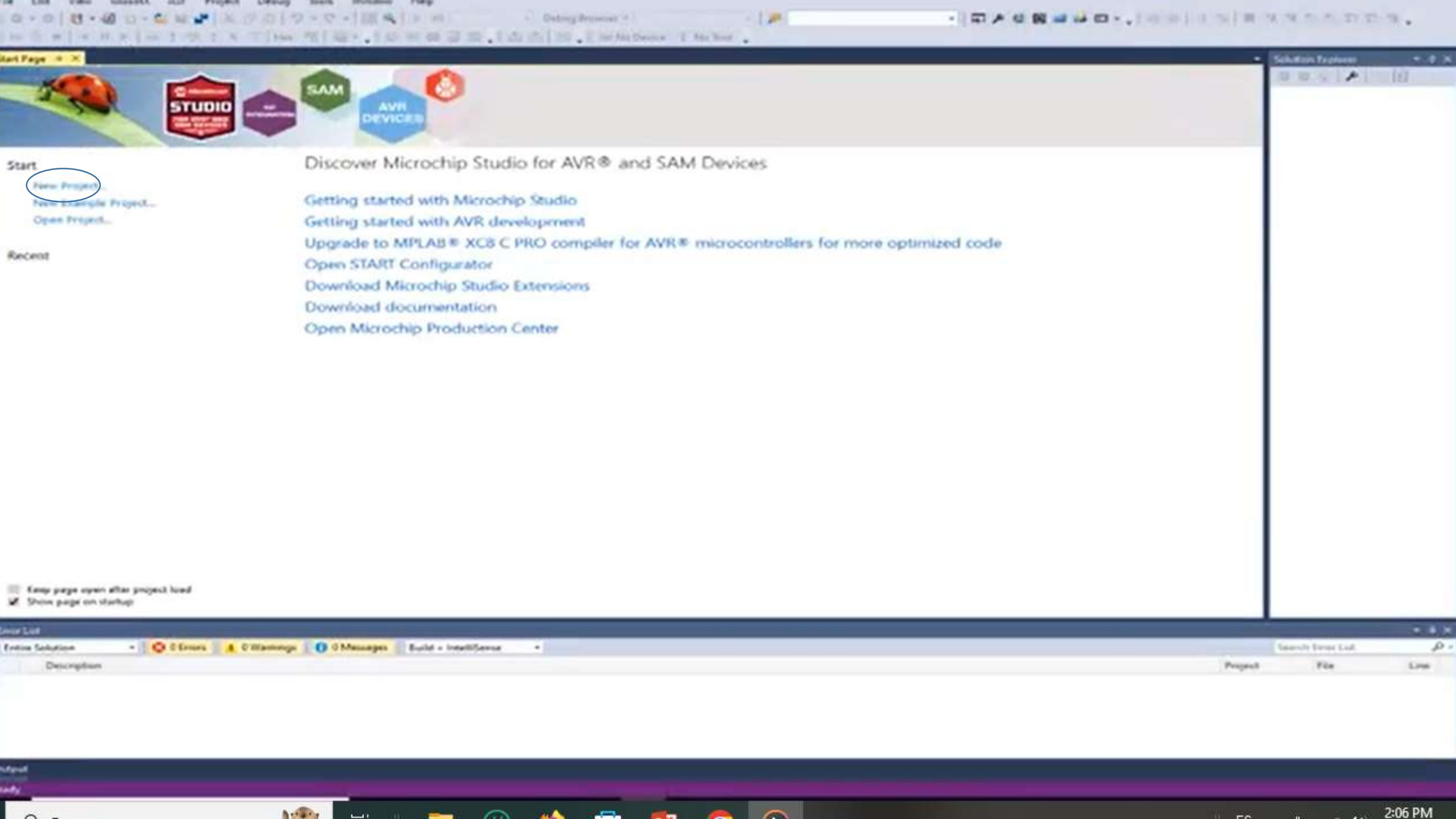
Configurar Puerto B como salida y mostrar el valor 0X01, el pin PB0 debe verse activado.



Ejemplo:

Configurar Puerto B como salida y mostrar el valor 0X01, el pin PB0 debe verse activado.





New Project

Recent

Installed

C/C++
Assembler
Microchip Studio Solution


Sort by: Default

AVR Assembler Project

Assembler

Search Installed Templates (Ctrl+E)

Type: Assembler
Creates an AVR 8-bit Assembler project



Name: Puertos

Location: C:\

Solution name: Puertos

Browse...

☒ Create directory for solution

OK

Cancel

Discover Microchip Studio for AVR® and SAM Devices

Getting started

Getting started

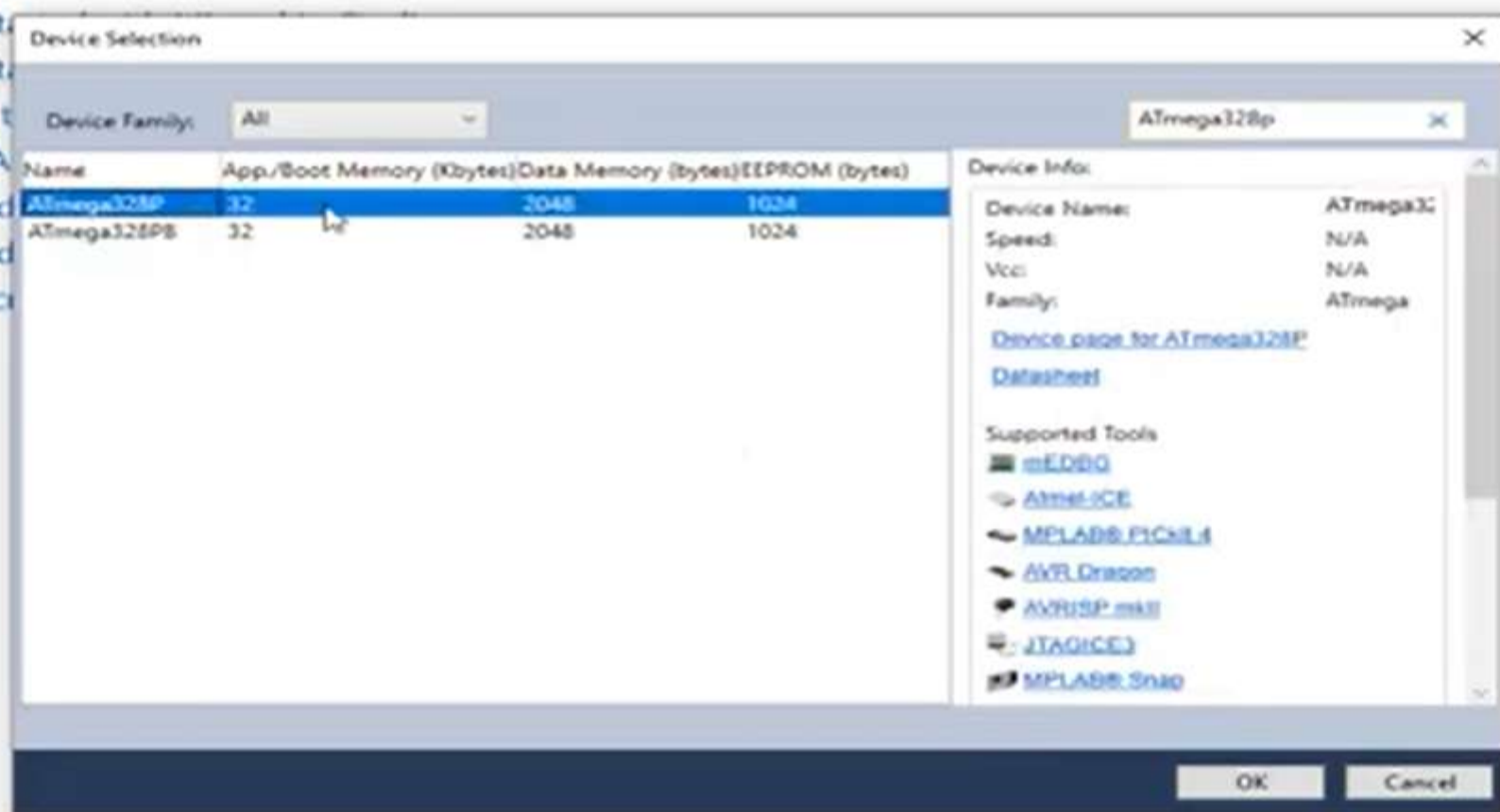
Upgrade to

Open STA

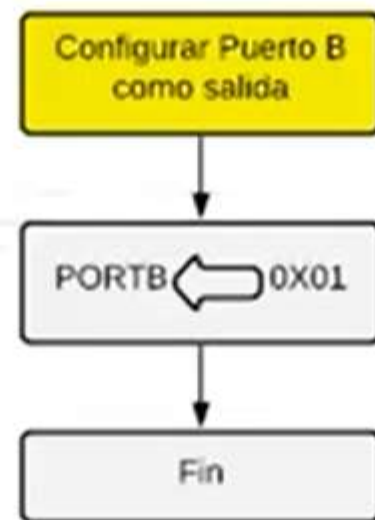
Download

Download


Open Mic

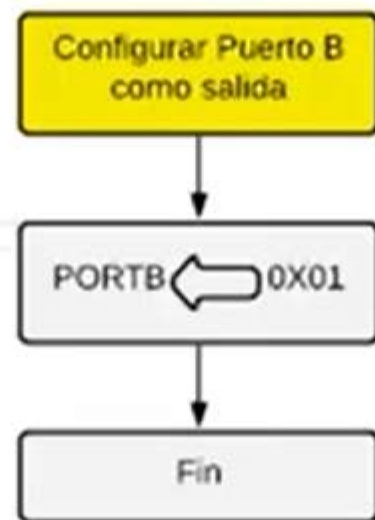


ct load

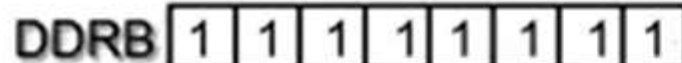
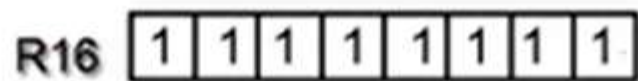


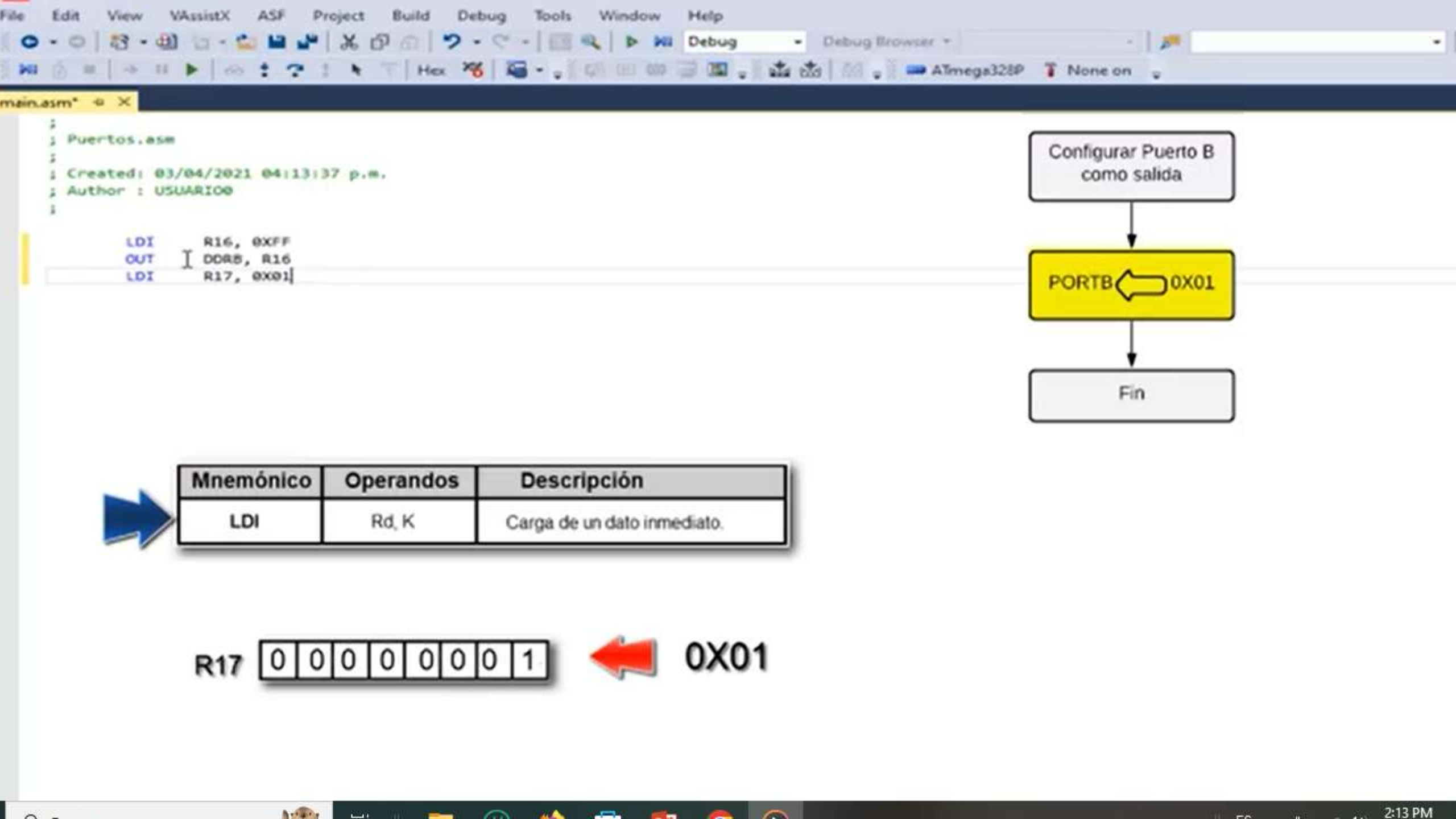
Mnemónico	Operandos	Descripción
LDI	Rd, K	Carga de un dato inmediato.

R16 1 1 1 1 1 1 1 1  0xFF



Mnemónico	Operandos	Descripción
OUT	P, Rr	Cargar en I/O un registro de trabajo





```
;
; Puertos.asm
;
; Created: 03/04/2021 04:13:37 p.m.
; Author : USUARIO0
;
```

```
LDI R16, 0xFF
OUT DDRA, R16
LDI R17, 0x01
```

Configurar Puerto B
como salida

PORTB ← 0x01

Fin

Mnemónico	Operandos	Descripción
LDI	Rd, K	Carga de un dato inmediato.

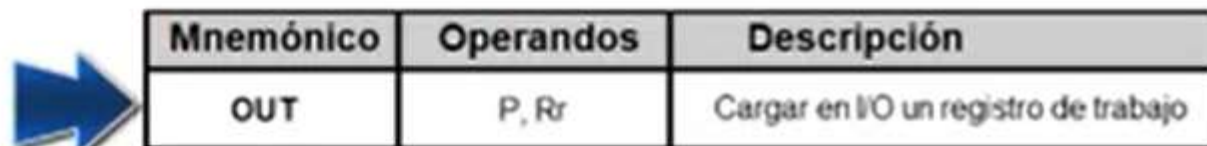
R17 0 0 0 0 0 0 0 1 ← 0x01

```

;
; Puertos.asm
;
; Created: 03/04/2021 04:13:37 p.m.
; Author : USUARIO
;

```

```
LDI    R16, 0xFF
OUT    I_DDRB, R16
LDI    R17, 0x01
OUT    PORTB, R17
```



2:14 PM

FileEditViewVAssistXASFProjectBuildDebugToolsWindowHelp

Debug

Debug Browser

ATmega328PNone on

main.asm

```
1
2 Puertos.asm
3
4 Created: 03/04/2021 04:13:37 p.m.
5 Author : USUARIO0
6
7
8 LDI R16, 0xFF
9 OUT DDRC, R16
10 LDI R17, 0x01
11 OUT PORTB, R17
```

Configurar Puerto B como salida

PORTB ← 0x01

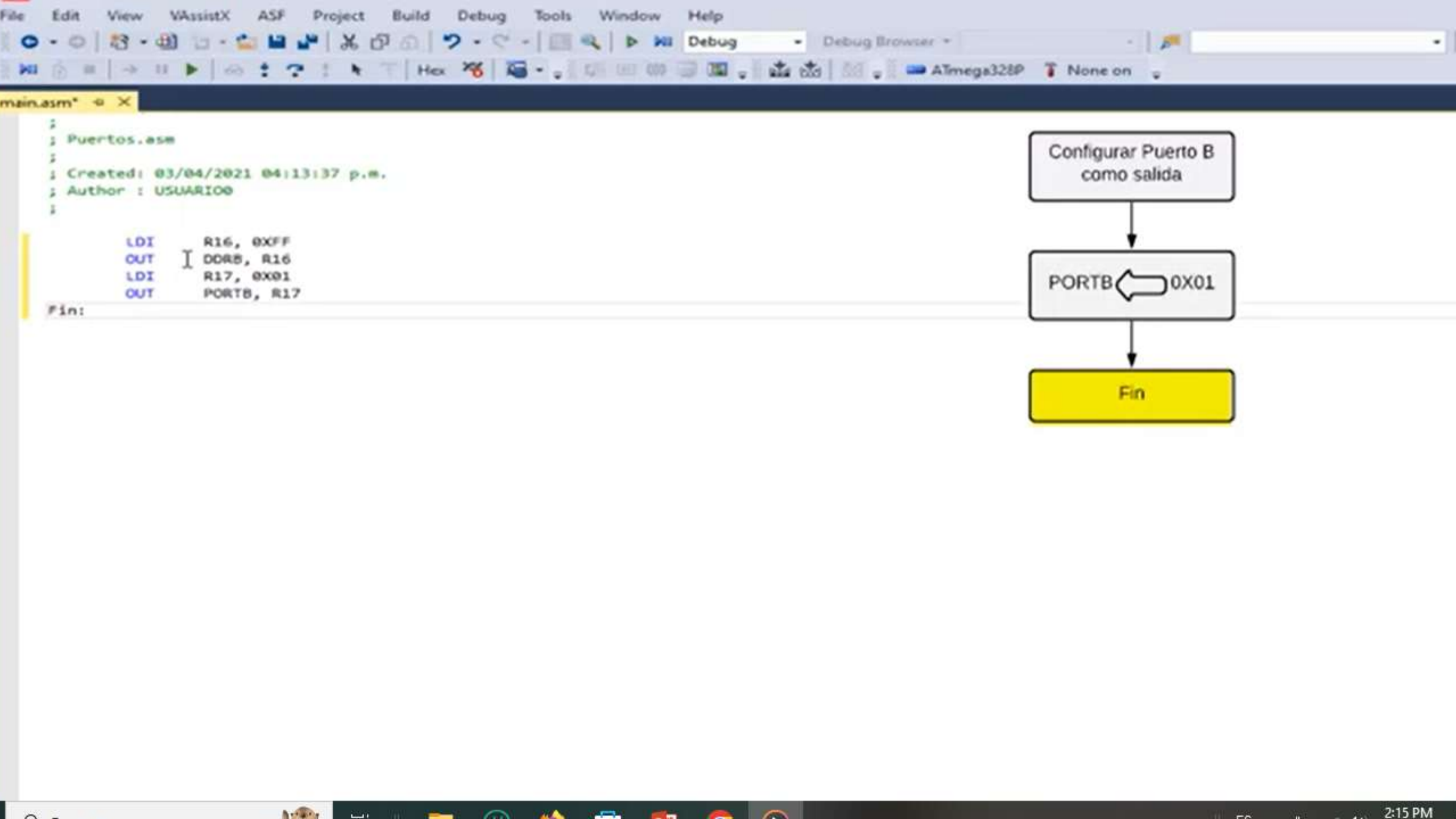
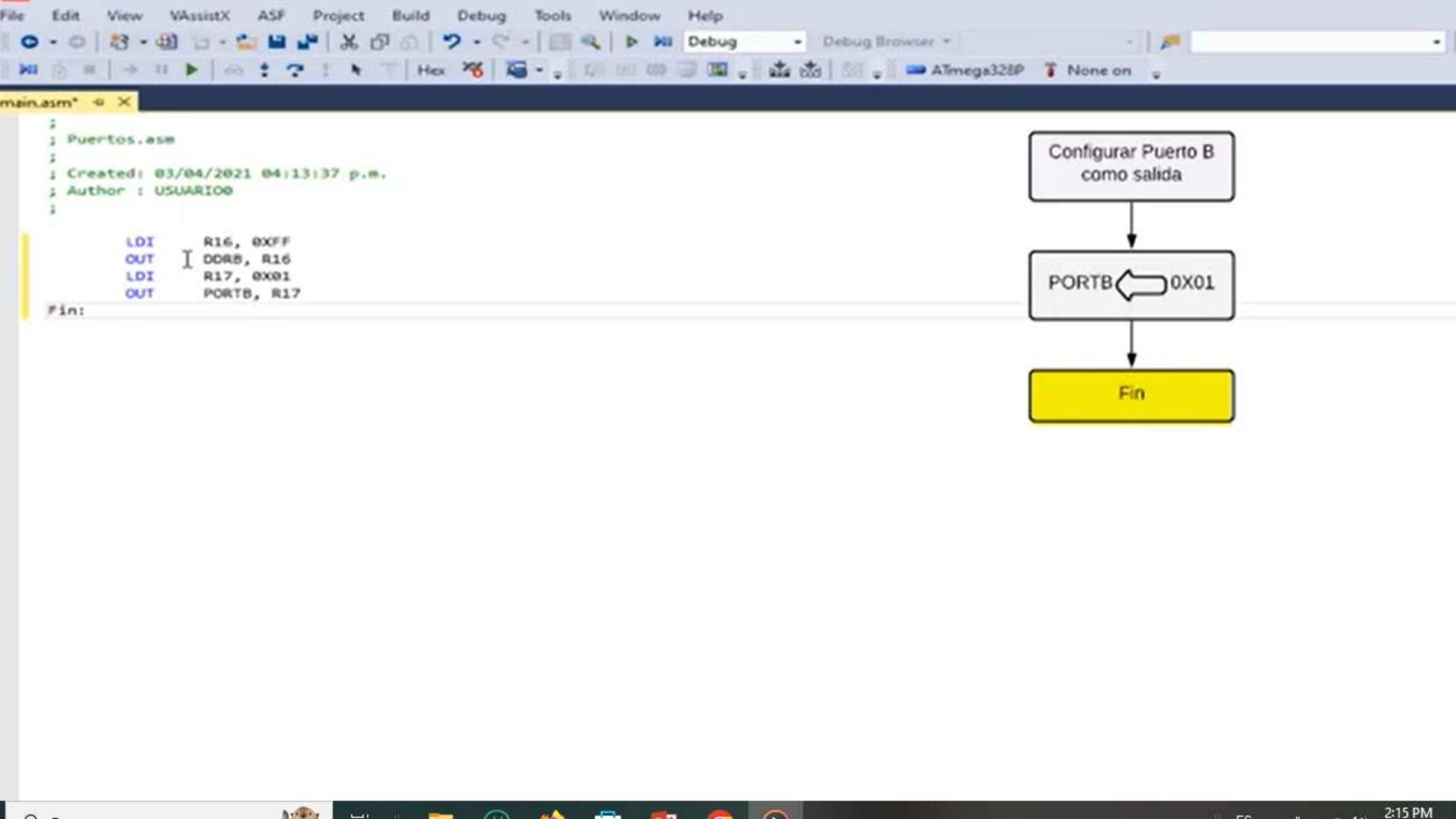
Fin

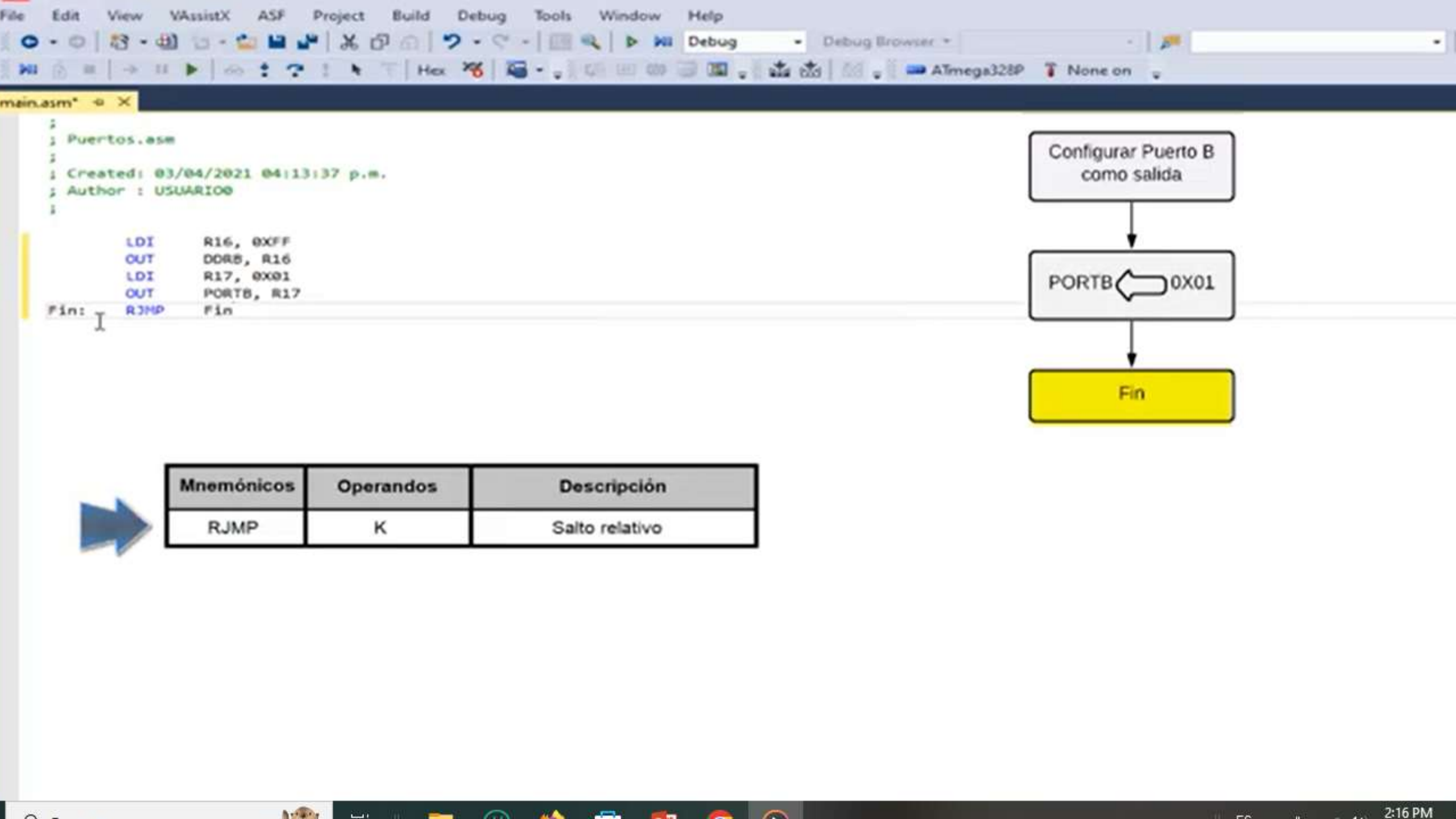
Etiqueta

Direcciones de salto

Nombres de subrutina

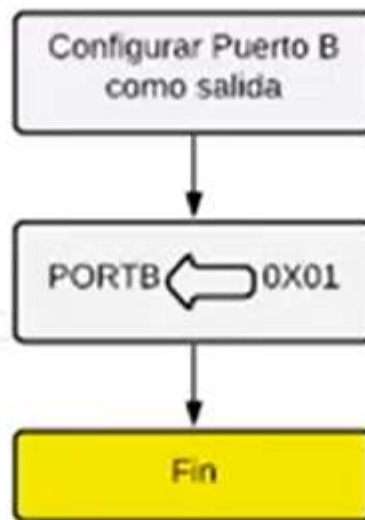
2:14 PM



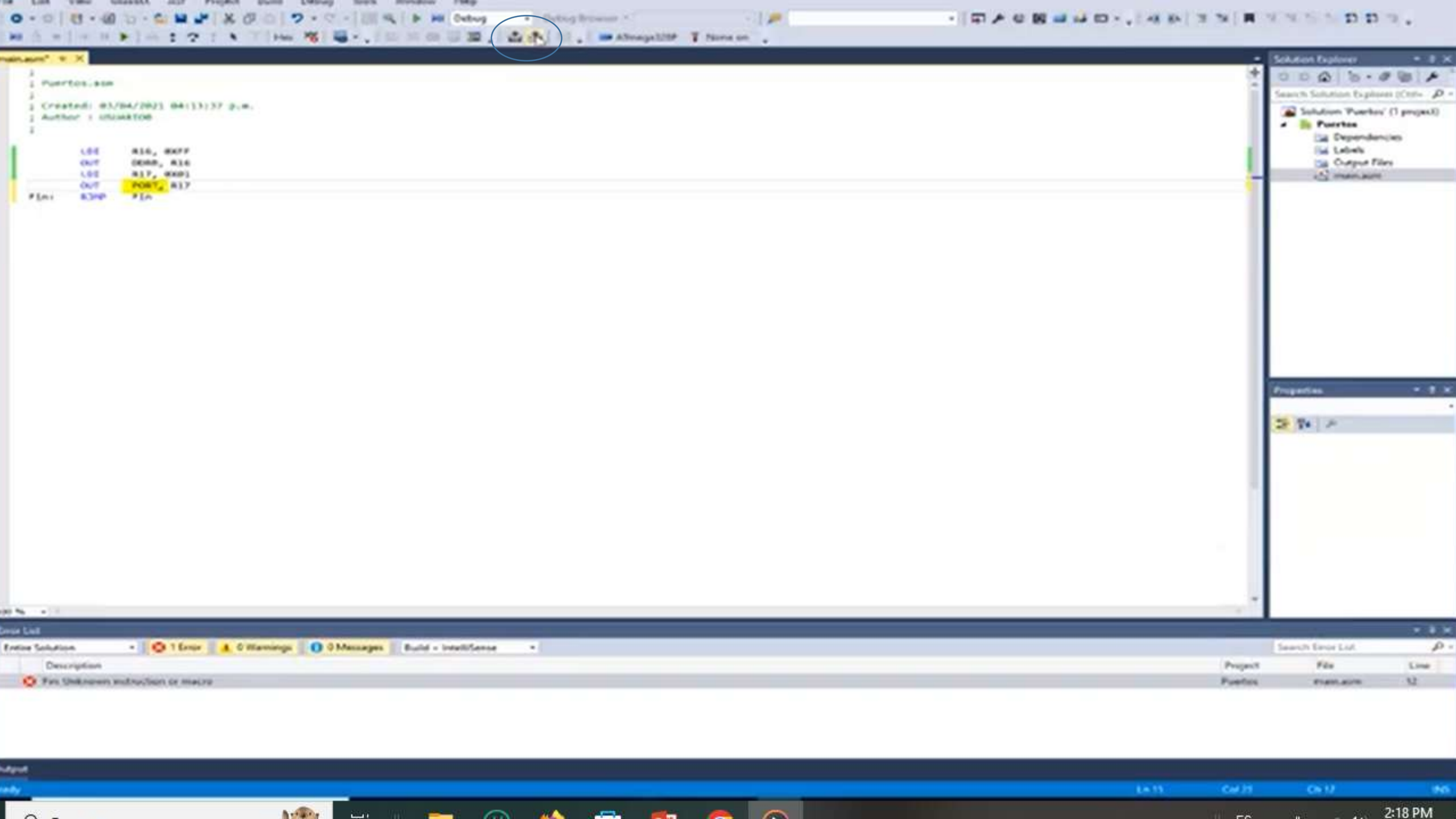


```
;
; Puertos.asm
;
; Created: 03/04/2021 04:13:37 p.m.
; Author : USUARIO00
;
```

```
LDI    R16, 0XFF
OUT    DDRB, R16
LDI    R17, 0X01
OUT    PORTB, R17
Fin:   RJMP  Fin
```



Mnemónicos	Operandos	Descripción
RJMP	K	Salto relativo



```
LDI R17, 0x001
OUT PORTB, R17
FIN: Rjmp Fin
```

Output

Show output from: Build

Segment	Address	Start	End	Size	Used	Free	Used %
[.text]	0x000000	0x000000	10	0	10	32768	0.0%
[.data]	0x000100	0x000100	0	0	0	2048	0.0%
[.bss]	0x000000	0x000000	0	0	0	1024	0.0%

Assembly complete, 0 errors, 0 warnings

Done executing task "RunAssemblerTask".

Done building target "CoreBuild" in project "Puentes.asmproj".

Target "PostBuildEvent" skipped, due to false condition; ('\$(PostBuildEvent)' != '') was evaluated as ('' != '').

Target "Build" in file "C:\Program Files\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Puentes\Puentes.asmproj" (entry point):

Done building target "Build" in project "Puentes.asmproj".

Done building project "Puentes.asmproj".

Build succeeded.

***** Build: 1 succeeded or up-to-date, 0 failed, 0 skipped *****

Output

Build succeeded

Explicación

LDI R16, 0xFF ; Cargar 0xFF (11111111 en binario) en el registro R16.

OUT DDRB, R16 ; Configurar todos los pines del puerto B como salidas.

LDI R17, 0x01 ; Cargar 0x01 (00000001 en binario) en el registro R17

OUT PORTB, R17 ; Encender solo el pin 0 del puerto B (PB0)

Fin: RJMP Fin ; Bucle infinito, salta siempre a la etiqueta Fin