

Εισαγωγή στην AVR-C

Παρουσιαστής: Χριστιανίδης Βασίλειος

Brought you by: <https://piday.gr/>

<https://gitlab.com/Basilisvirus/atmel-studio-examples-atmega328p-avrc/tree/master>



Ποιο πρόβλημα λύνει η Avr-C?

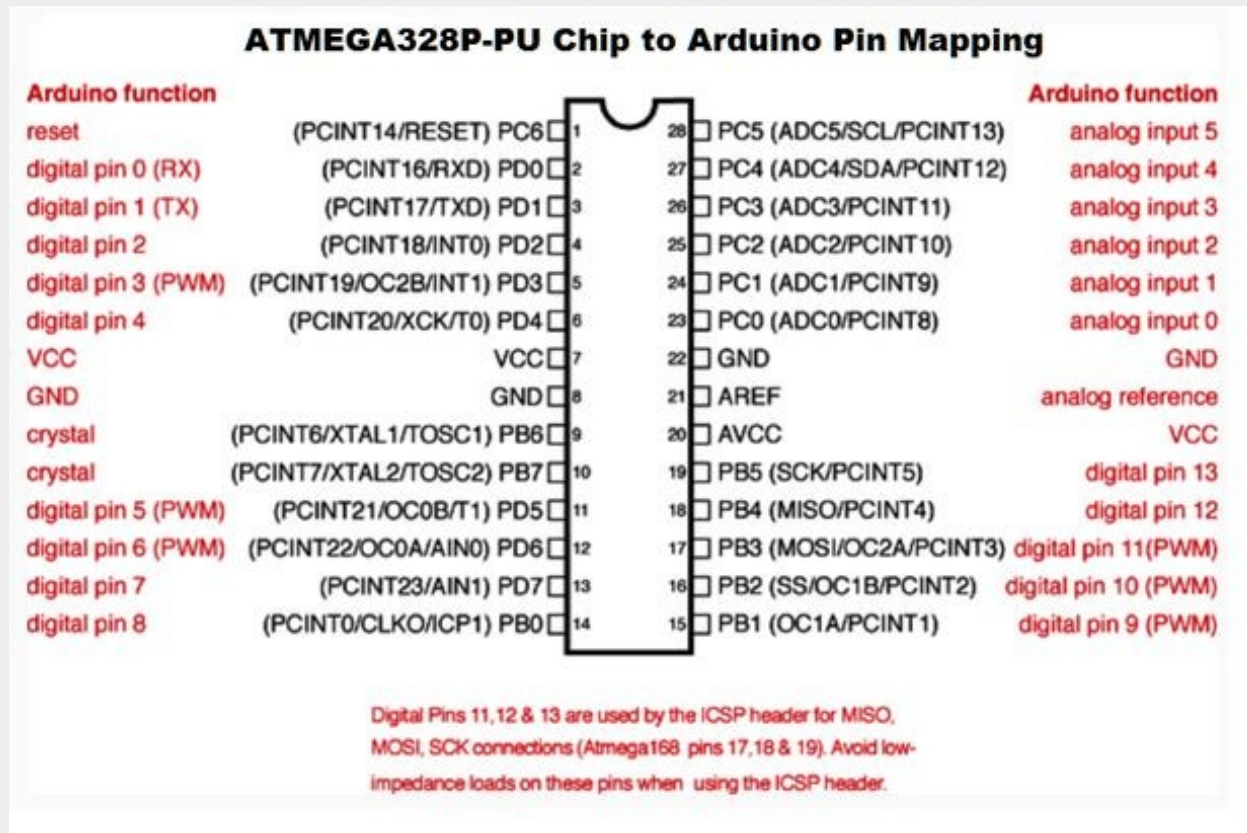
Γράφοντας τα προγράμματα των μικροελεγκτών της atmel σε avr-c αντί σε C++ και βιβλιοθηκών arduino, επιτυγχάνονται τα εξής:

- Γρηγορότερη εκτέλεση του προγράμματος του μικροελεγκτή, χρήση όλων των δυνατοτήτων του μικροελεγκτή.
- Καλύτερος έλεγχος του προγράμματος και ευκολότερο debugging.
- Εκμάθηση της λειτουργίας του μικροελεγκτή από τον χρήστη
- Ευκολότερη χρήση νέων μικροελεγκτών οι οποίοι δε διαθέτουν έτοιμες βιβλιοθήκες και community
- Και είναι μια επαγγελματική προσέγγιση στην ανάπτυξη ενός έργου, έναντι της χρήσης Arduino.

Βήματα για να ξεκινήσω AVR-C

Η Pi-Day αναγνώρισε τη σχετική δυσκολία εκκίνησης εκμάθησης της AVR-C και δημιούργησε το repository στο GitLab, όπου διατυπώνουμε όλα όσα μάθαμε.

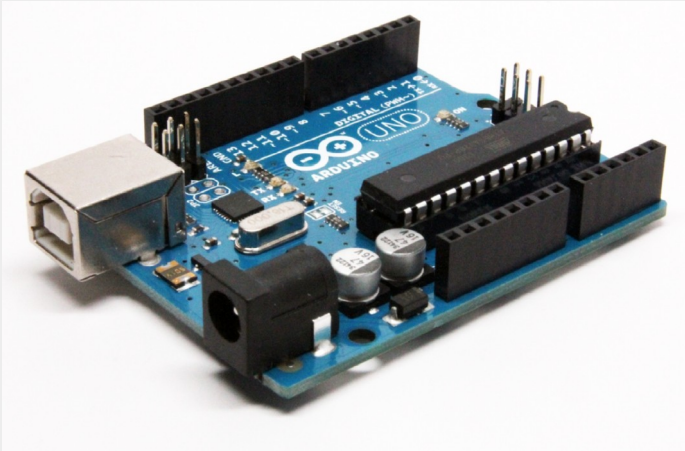
Θα χρειαστεί η παρακάτω εικόνα, για reference στα pins. Με **κόκκινο**, είναι οι ονομασίες των pins για το **arduino**, με **μαύρο** οι ονομασίες για AVR-C.



Θα χρειαστεί επίσης το excel **atmel-studio-examples-atmega328p-avrc\1Atmega328p map & datasheet\Atmega328.ods** από το repo:

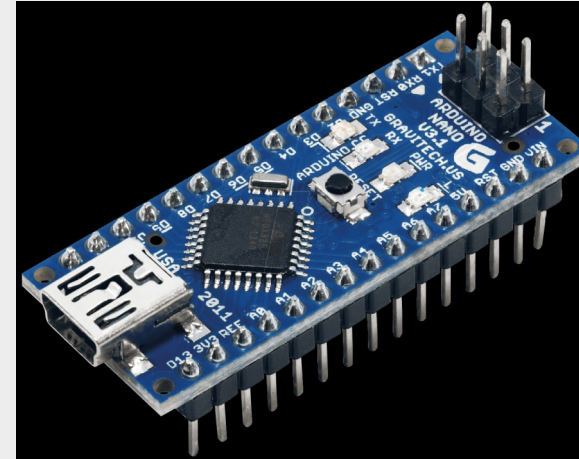
Gitlab Repo: <https://gitlab.com/Basilisvirus/atmel-studio-examples-atmega328p-avrc>

Εργαλεία

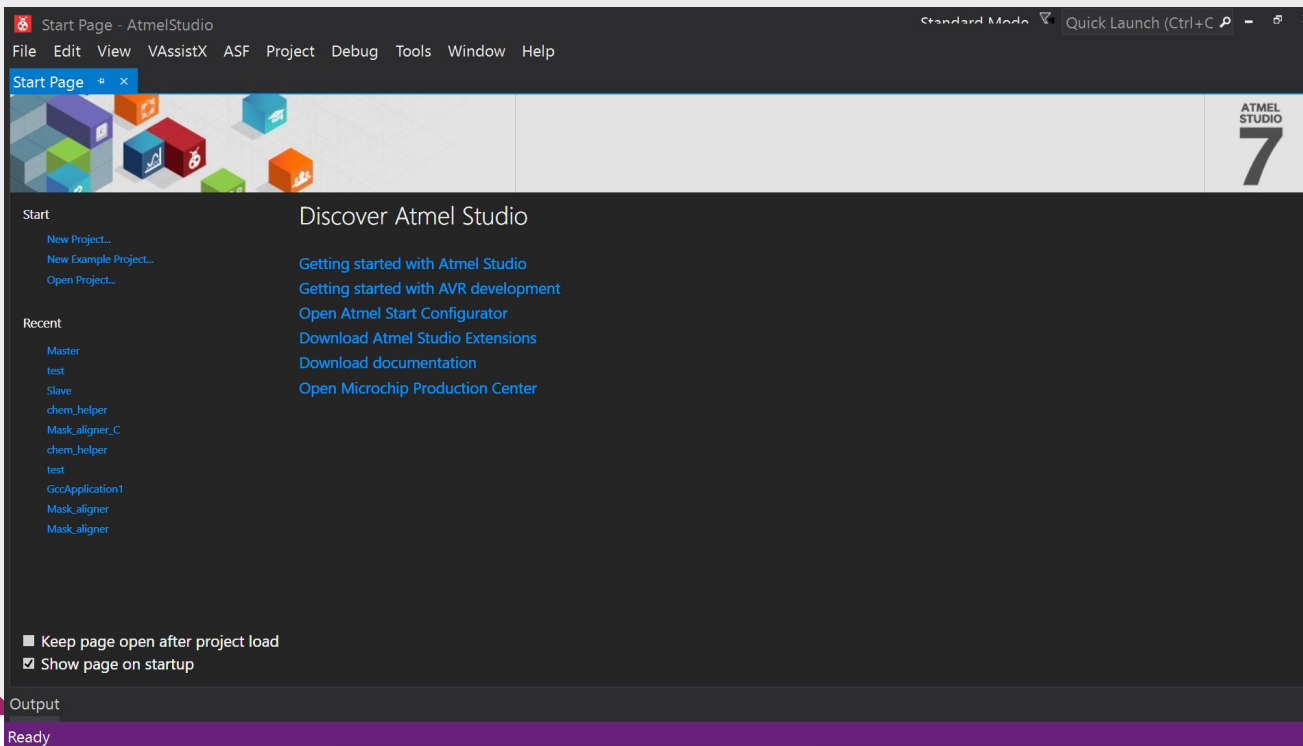


Arduino uno

ή

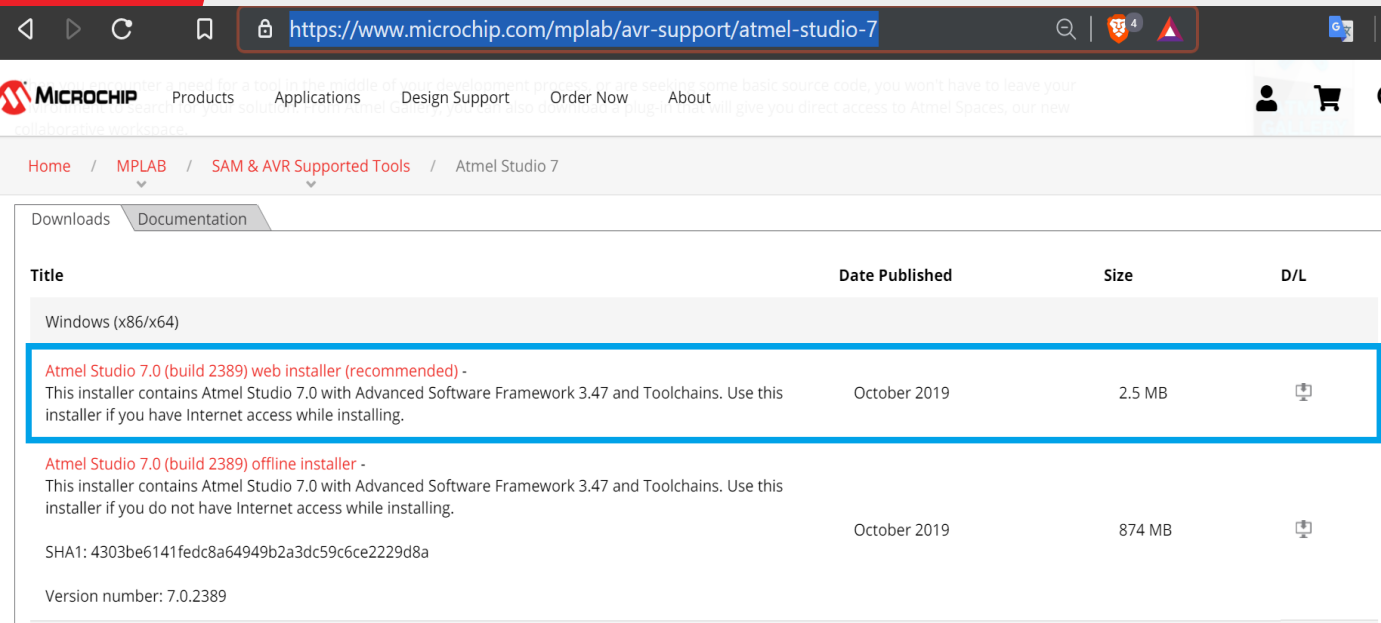


Arduino nano



Και Atmel Studio, το επίσημο πρόγραμμα για avr-c της atmel.

Εγκατάσταση atmel studio



Microchip Products Applications Design Support Order Now About

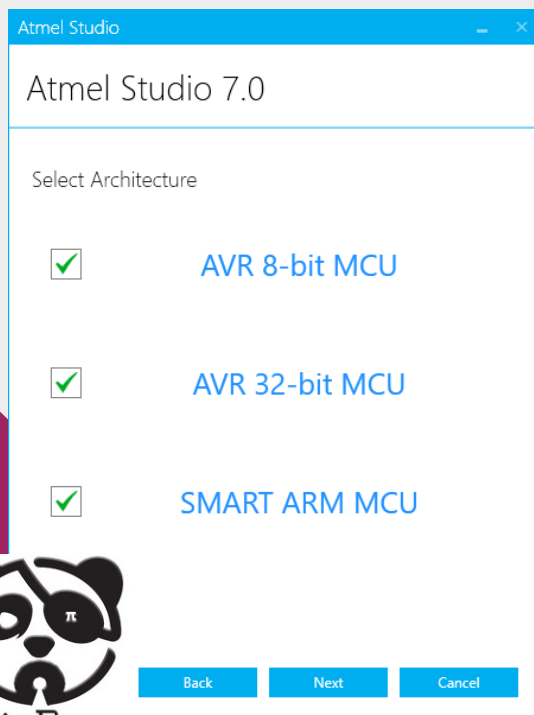
Home / MPLAB / SAM & AVR Supported Tools / Atmel Studio 7

Downloads Documentation

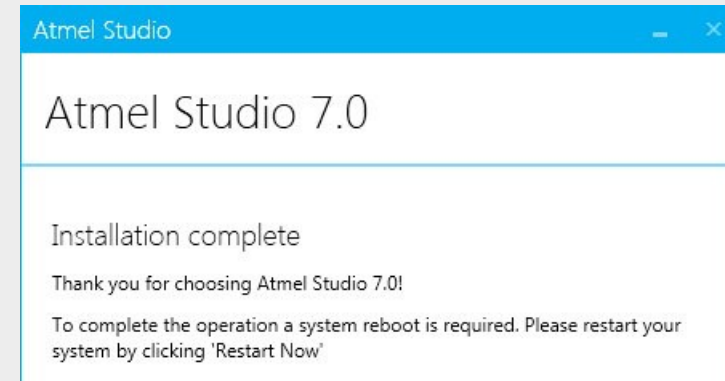
Title	Date Published	Size	D/L
Windows (x86/x64)			
Atmel Studio 7.0 (build 2389) web installer (recommended) - This installer contains Atmel Studio 7.0 with Advanced Software Framework 3.47 and Toolchains. Use this installer if you have Internet access while installing.	October 2019	2.5 MB	
Atmel Studio 7.0 (build 2389) offline installer - This installer contains Atmel Studio 7.0 with Advanced Software Framework 3.47 and Toolchains. Use this installer if you do not have Internet access while installing.	October 2019	874 MB	
SHA1: 4303be6141fedc8a64949b2a3dc59c6ce2229d8a			
Version number: 7.0.2389			

1) κατεβάζουμε το αρχείο, μόνο για windows

2) τικάρουμε όλες τις επιλογές



3) Θα χρειαστεί restart όπου θα συνεχιστεί η εγκατάσταση



4) κάνουμε όσα updates ζητηθούν μετα το install

<https://wspublishing.net/avr-c/installing-atmel-studio-7/>



Το πρώτο πρόγραμμα, led blink

Start Page - AtmelStudio

Standard Mode Quick Launch (Ctrl+Q)

File Edit View VAssistX ASF Project Debug Tools Window Help

Start Page

New Project

Recent

Installed

C/C++

Assembler

AtmelStudio Solution

Sort by: Default

Search Installed Templates (Ctrl+E)

Type: C/C++

Creates an AVR 8-bit or AVR/ARM 32-bit C project

3

1

2

4

Name: GccApplication1

Location: c:\users\basilisvirus\Documents\Atmel Studio\7.0

Solution name: GccApplication1

Browse...

Create directory for solution

OK Cancel

ATMEL STUDIO

Debug Browser

No Device No Tool

Start

New Project...

New Example Project...

Open Project...

Recent

GCC C ASF Board Project C/C++

GCC C Executable Project C/C++

GCC C Static Library Project C/C++

GCC C++ Executable Project C/C++

GCC C++ Static Library Project C/C++

SAM L11 Secure Solution v1.0 C/C++

Create project from Arduino sketch C/C++

```
#include <avr/io.h>
int main(void)
{
    printf("Hello")
}
```

GCC



Το πρώτο πρόγραμμα, led blink

Start Page - AtmelStudio

Standard Mode Quick Launch

File Edit View VAssistX ASF Project Debug Tools Window Help

Start Page

ATMEL STUDIO 7

Start

- New Project...
- New Example Project...
- Open Project...

Recent

Device Selection

Device Family: All

328p 1

Name	App./Boot Memory (Kbytes)	Data Memory (bytes)	EEPROM (bytes)
ATmega328P	32	2048	1024
ATmega328PB	32	2048	1024

2

Device Info:

Device Name: ATmega328P

Speed: N/A

Vcc: N/A

Family: ATmega

[Device page for ATmega328P](#)

[Datasheet](#)

Supported Tools

- Atmel-ICE
- AVR Dragon
- AVRISP mkII
- AVR ONE!
- EDBG
- EDBG MSD
- JTAGICE3

OK Cancel

3



Ο κώδικας

```
main.c X ASF Wizard
main.while while (1)

#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    while (1)
    {
        // Empty line
    }
}
```

Πρώτα πρέπει να ενημερωθεί ο μικροελεγκτής για τη ταχύτητα του XTAL.

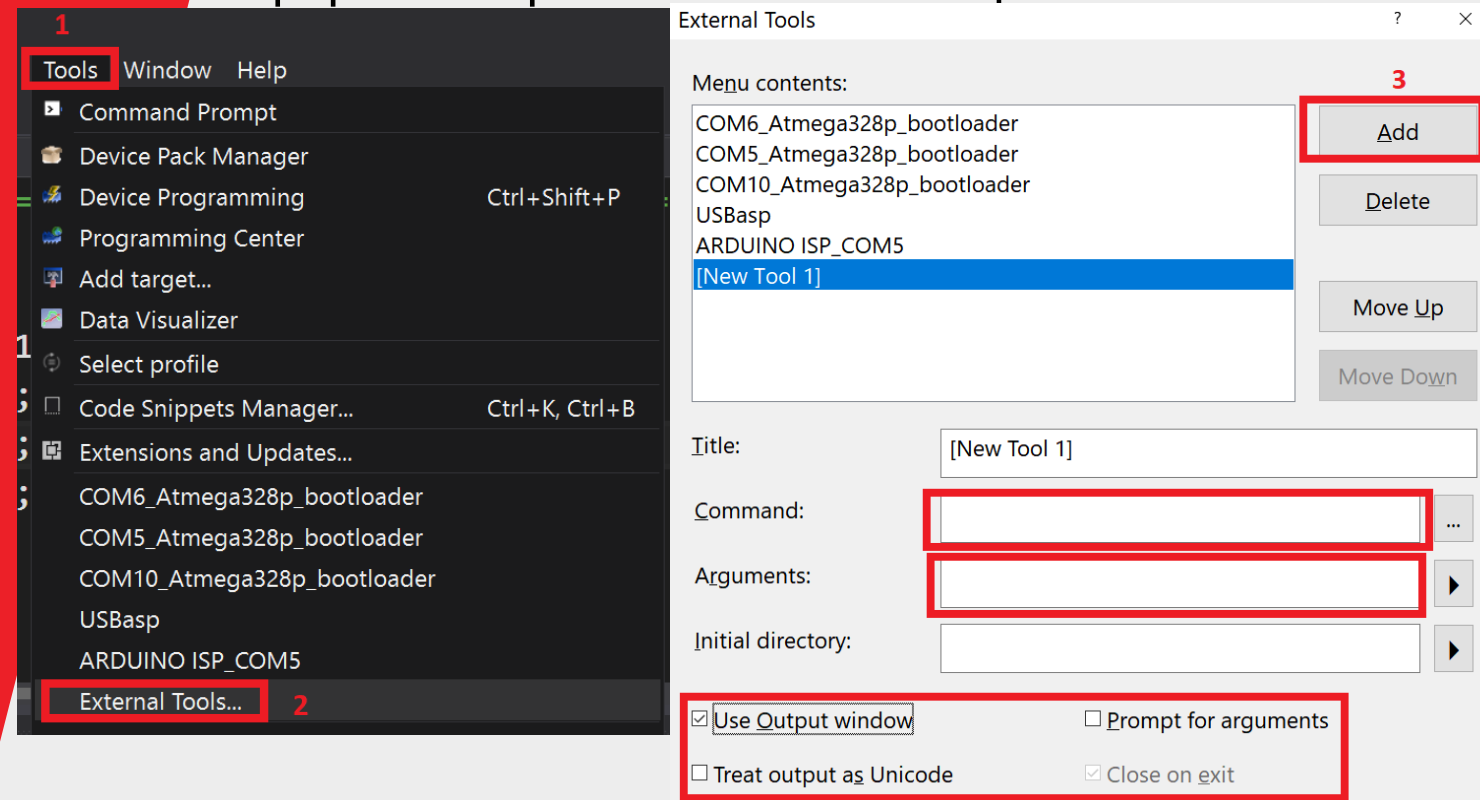
Ύστερα, εισάγονται οι βιβλιοθήκες προς χρήση. Η `io.h` είναι η βασική βιβλιοθήκη για χρήση των pins του μικροελεγκτή.

Η `main` είναι η πρώτη συνάρτηση που θα εκτελεστεί, και η `while(1)` η συνθήκη που επαναλαμβάνεται συνεχώς.



Ανέβασμα κώδικα

Για να φορτώσουμε τον κώδικα στη πλακέτα:



Title: το όνομα του tool, ότι όνομα βάλεις, να το θυμάσαι

Αφου ακολουθήσεις τα βήματα της διαφάνειας, Πήγαινε tools/(το tool που έφτιαξες) για να δεις αν θα ανέβει επιτυχώς το πρόγραμμά σου

Command: **C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avrdude.exe**

Arguments: **-C"C:\Program Files (x86)\Arduino\hardware\tools\avr\etc\avrdude.conf" -v -patmega328p -carduino -P**COM6** -b115200 -D -Uflash:w:"\$(ProjectDir)Debug\$(TargetName).hex":i**

Μόνο το COM6 θα χρειαστεί να αλλαχθεί, ανάλογα με το com που είναι συνδεδεμένο το arduino. (Μπορείς να το δεις από το Arduino IDE).



Επεξηγηματικό βιντεο:<https://www.youtube.com/watch?v=zEbSQaQJvHI>

Ο κώδικας

Έστω ότι θέλουμε να χρησιμοποιήσουμε το pin 10 του arduino. Σύμφωνα με το map, αυτό είναι το PB2 pin. Ανήκει στο Port B, αριθμός 2.

Pin to Arduino Pin Mapping	
	Arduino function
28 □ PC5 (ADC5/SCL/PCINT13)	analog input 5
27 □ PC4 (ADC4/SDA/PCINT12)	analog input 4
26 □ PC3 (ADC3/PCINT11)	analog input 3
25 □ PC2 (ADC2/PCINT10)	analog input 2
24 □ PC1 (ADC1/PCINT9)	analog input 1
23 □ PC0 (ADC0/PCINT8)	analog input 0
22 □ GND	GND
21 □ AREF	analog reference
20 □ AVCC	VCC
19 □ PB5 (SCK/PCINT5)	digital pin 13
18 □ PB4 (MISO/PCINT4)	digital pin 12
17 □ PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
16 □ PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
15 □ PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

used by the ICSP header for MISO, pins 17, 18 & 19). Avoid low-voltage pins when using the ICSP header.

Άρα, από το excel, θα χρειαστούν τα Registers **DDRB** (bit **DDB2**) για να θέσουμε το pin 10 ως OUTPUT και ύστερα το Register **PORTB** (**PORTB2**) για να γίνει HIGH ή LOW η έξοδος του pin 10.

DDB7 DDB6 DDB5 DDB4 DDB3 DDB2 DDB1 DDB0

To use a port as output, assign 1 to it

`DDRB |= 0B00100000;`

If a port is input, you can activate its pullup resistor
Using PORTB

PORTB7 PORTB6 PORTB5 PORTB4 PORTB3 PORTB2 PORTB1 PORTB0

First, set PORT input or output using DDRx

By setting a bit of PORT to 1:

if PORT is output, the output bit is set to HIGH.

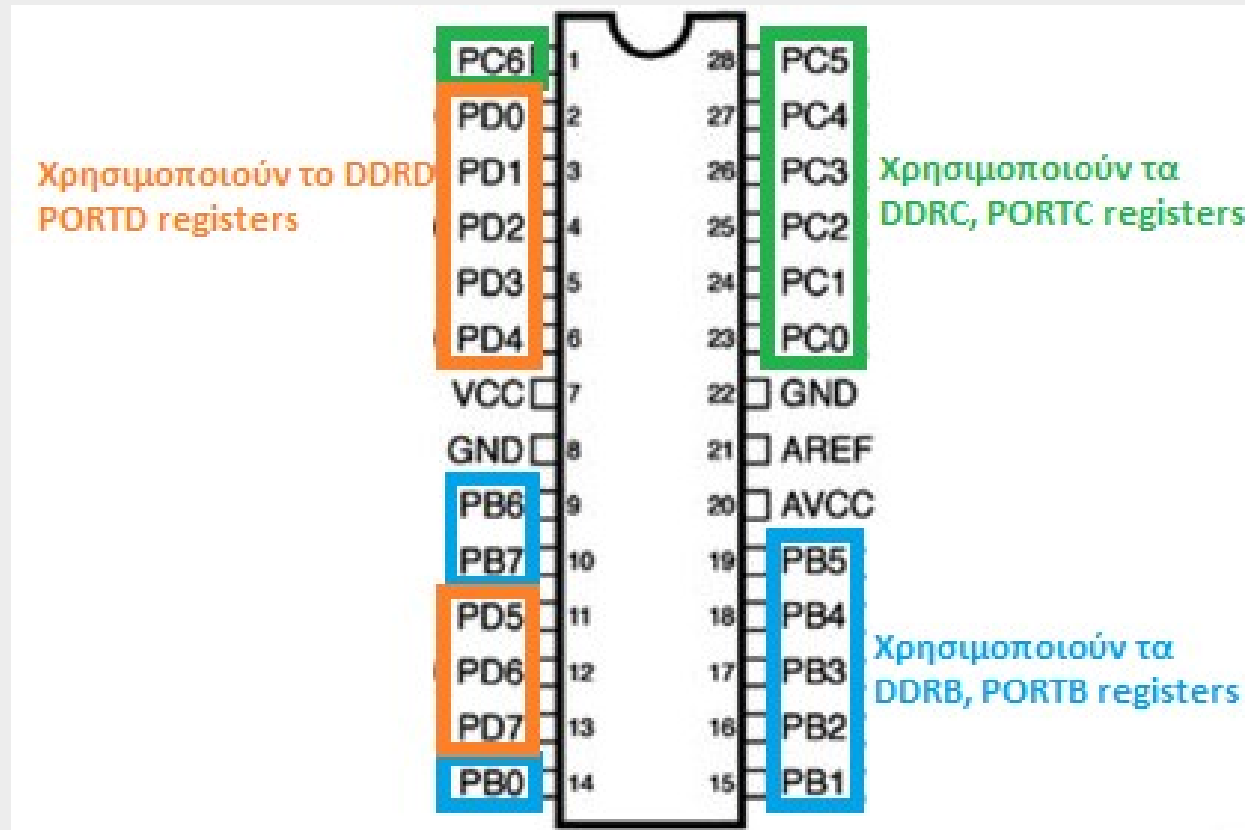
if PORT is input, the internal pull-up resistor is activated

`PORTB |= 0B00100000;`

x5H	x4H
<u>xDH</u>	<u>xCH</u>
R5	R4
R13	R12
R21	R20
R29	R28
<u>PORTB</u>	<u>DDRB</u>
-	-



Ο κώδικας



17H	R23	R22	R21	R20	R19	R18	R17	R16	1
1FH	R31	R30	R29	R28	R27	R26	R25	R24	1
27H	<u>DDRC</u>	<u>PINC</u>	<u>PORTB</u>	<u>DDRB</u>	<u>PINB</u>	—	—	—	2
2FH	—	—	—	—	<u>PORTD</u>	<u>DDRD</u>	<u>PIND</u>	<u>PORTC</u>	2
37H	<u>TIFR2</u>	<u>TIFR1</u>	<u>TIFR0</u>	—	—	—	—	—	3

Ο κώδικας

Έστω ότι θέλουμε να χρησιμοποιήσουμε το pin 10 του arduino. Σύμφωνα με το map, αυτό είναι το PB2 pin. Ανήκει στο Port B, αριθμός 2.

Pin to Arduino Pin Mapping	
	Arduino function
28 □ PC5 (ADC5/SCL/PCINT13)	analog input 5
27 □ PC4 (ADC4/SDA/PCINT12)	analog input 4
26 □ PC3 (ADC3/PCINT11)	analog input 3
25 □ PC2 (ADC2/PCINT10)	analog input 2
24 □ PC1 (ADC1/PCINT9)	analog input 1
23 □ PC0 (ADC0/PCINT8)	analog input 0
22 □ GND	GND
21 □ AREF	analog reference
20 □ AVCC	VCC
19 □ PB5 (SCK/PCINT5)	digital pin 13
18 □ PB4 (MISO/PCINT4)	digital pin 12
17 □ PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
16 □ PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
15 □ PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

used by the ICSP header for MISO, negative pins 17, 18 & 19). Avoid low-voltage pins when using the ICSP header.

Άρα, από το excel, θα χρειαστούν τα Registers **DDRB** (bit **DDB2**) για να θέσουμε το pin 10 ως OUTPUT και ύστερα το Register **PORTB (PORTB2)** για να γίνει HIGH ή LOW η έξοδος του pin 10.

```
#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    //declare PORTB2 as output
    DDRB = 0B00000100;

    while (1) //Ενώ (ισχύει η συνθήκη) κάνε...
    {
        PORTB = 0B00000100; //light the led
        _delay_ms(1000); //wait 1 sec
        PORTB = 0B00000000; //turn it off
        _delay_ms(1000); //wait 1 sec
    }
}
```



Ο κώδικας - Σύγκριση με arduino

- Στην AVR-C πρέπει να φτιάξεις μόνοις την loop() με while (1).
- Τα pinMode() δεν είναι τόσο 'φιλικά' προς τον χρήστη, αλλά δεν φορτώνει το arduino με βαριές βιβλιοθήκες. (πίσω από το pinMode() τρέχουν πολλές γραμμές κώδικα).
- Στην AVR-C βάζεις μόνοις σου τις βιβλιοθήκες που χρειάζεσαι και δηλώνεις τη ταχύτητα του επεξεργαστή (F_CPU)

```
#define F_CPU 16000000UL
```

```
#include <avr/io.h>
```

```
#include <util/delay.h>
```

```
int main(void)
```

```
{
```

```
    //declare PORTB2 as output
```

```
    DDRB = 0B00000100;
```

```
    while (1) //Ενώ (ισχύει η συνθήκη) κάνε...
```

```
    {
```

```
        PORTB = 0B00000100; //light the led
```

```
        _delay_ms(1000); //wait 1 sec
```

```
        PORTB = 0B00000000; //turn it off
```

```
        _delay_ms(1000); //wait 1 sec
```

```
int led = 10;
```

```
void setup() {
```

```
    pinMode(led, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    digitalWrite(led, HIGH);
```

```
    delay(1000);
```

```
    digitalWrite(led, LOW);
```

```
    delay(1000);
```

```
}
```



Διαφορετική προσέγγιση

Επειδή στο προηγούμενο κώδικα επηρεάζαμε και όλα τα port της PORTB, όταν βάζαμε μία τιμή σε αυτή, μπορούμε να πειράξουμε μόνο το bit που θέλουμε με τις λογικές πράξεις:

Μπορώ να χρησιμοποιήσω `Roll Left (1 << PORTB2)` //Κάνε 1 το PORTB2
ή να χρησιμοποιήσω μάσκα `&= 0B11111011` // Κάνε 0 το PORTB2

Προτιμάται η χρήση της μάσκας όταν χρησιμοποιούμε λογικό `&=`

```
#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    //declare PORTB2 as output
    DDRB = 0B00000100;

    while (1) //Ενώ (ισχύει η συνθήκη) κάνε...
    {
        PORTB |= (1 << PORTB2); //light the led
        _delay_ms(1000); //wait 1 sec
        PORTB &= 0B11111011; //turn it off
        _delay_ms(1000); //wait 1 sec
    }
}
```



Roll Left

Συντακτικό:

Byte |= (αριθμός << θέσεις_μετακίνησης)

^
|

Λογική πράξη

Άσκηση: θέλω στο PORTB να βάλω το '1' στο τρίτο του bit [2]

Δηλαδή, έστω το PORTB είναι 0000 0000, εγώ θα το κάνω 0000 0100

Μπορώ να κάνω:

`PORTB |= (1 << 2);` //θα βάλει το '1' στο PORTB[0] και τα το μετακινήσει αριστερά 2 θέσεις, μέχρι να φτάσει στο PORTB[2].

-Το ίδιο κάνει και το `PORTB |= (1 << PORTB2)`, για ευκολία

Άρα θα καταλήξει να κάνει `PORTB |= 0000 0100;` // το αναλυω στην επόμενη διαφάνεια

PORTB

PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]

First, set PORT input or output using DDRx

By setting a bit of PORT to 1:
if PORT is output, the output bit is set to HIGH.
if PORT is input, the internal pull-up resistor is activated

```
PORTB |= 0B00100000;
```

Λογικές πράξεις

Ουσιαστικά το Roll Left φτιάχνει μία Μάσκα και ύστερα εκτελεί τη λογική πράξη, όπως είδαμε στη προηγούμενη διαφάνεια.

Η εικόνα είναι από το [excel](#)

	AND (&)		OR ()		XOR (^)	
Μάσκα	1	0	1	0	1	0
αποτέλεσμα	αθηκτα	0	1	αθηκτα	αντιστροφή	αθηκτο
!!!κραταει τη τιμη του bit που θελω!!!						

Άσκηση: Να γίνει το τρίτο bit του PORTB μηδέν, έστω ότι το PORTB είναι 0000 0100. Θα το κάνω 0000 0000.

$\text{PORTB} \&= 1111\ 1011$ //σύμφωνα με τον πίνακα, βάζω '1' σε όποια bit θέλω να μείνουν άθικτα, και '0' σε όσα θέλω να γίνουν 0.



Μπορείτε να μάθετε να χρησιμοποιείτε την AVR-C χρησιμοποιώντας μόνο το διαδίκτυο (youtube, stack overflow, blogs) και το datasheet για τον κάθε μικροελεγκτή.

Το <https://www.avrfreaks.net/> avr freaks forum ειδικεύονται στους μικροελεγκτές AVR και μπορούν να λύσουν τις απορίες σας, εάν δεν βοηθηθείτε από κάποιο tutorial.

Η Π-Day διαθέτει προγραμματιστή avr και θα προσπαθήσει να λύσει τις απορίες σας.

Ευχαριστούμε.

Atmega328p παραδείγματα avr-c:

<https://gitlab.com/Basilisvirus/atmel-studio-examples-atmega328p-avrc>

Πως να χρησιμοποιήσεις το atmega328p χωρίς arduino:

<https://gitlab.com/Basilisvirus/atmega328p-standalone>



Our sponsors



www.pcbway.com

PCB design

PCB Assembly

Return and Refund

24 hours Customer Service

