Embedded Systems and IOT

2025/2026

Lecture 3

Assis. Prof. Dr. Elmahdy Maree

Embedded Systems and IOT

Arduino UNO

Embedded Systems and IOT

CH1: AVR microcontroller block diagram

Block Diagram of the AVR Architecture

Embedded Systems and IOT

Arduino Families

Embedded Systems and IOT

https://www.arduino.cc/en/hardware/#boards

For Professionals | For Education | For Makers | Products | Community | Documentation | SHOP

Nano Family Store Page

Arduino Nano 33 IoT	Arduino Nano RP2040 Connect	Arduino Nano ESP32	Arduino Nano 33 BLE Sense
Arduino Nano 33 BLE	Arduino Nano Every	Arduino Nano	Arduino Nano Motor Carrier
Nano Connector Carrier			

Embedded Systems and IOT

For Professionals | For Education | For Makers | Products | Community | Documentation | SHOP

MKR Family

Arduino MKR 1000 WiFi	Arduino MKR WiFi 1010	Arduino MKR FOX 1200
Arduino MKR WAN 1300	Arduino MKR WAN 1310	Arduino MKR GSM 1400
Arduino MKR NB 1500	Arduino MKR Vidor 4000	Arduino MKR Zero

Embedded Systems and IOT

For Professionals | For Education | For Makers | Products | Community | Documentation | SHOP

MKR Family

Shields

Arduino MKR ENV Shield Rev2	Arduino MKR GPS Shield	Arduino MKR IMU Shield
Arduino MKR RGB Shield	Arduino MKR Therm Shield	Arduino MKR ETH Shield
Arduino MKR 485 Shield	Arduino MKR CAN Shield	Arduino MKR MEM Shield

Embedded Systems and IOT

For Professionals | For Education | For Makers | Products | Community | Documentation | SHOP

UNO Family

Boards

Arduino UNO Q	Arduino UNO R4 WiFi	Arduino UNO R4 Minima	Arduino UNO R3
Arduino UNO Mini Limited Edition	Arduino UNO WiFi Rev2	Arduino UNO R3 SMD	

Embedded Systems and IOT

For Professionals | For Education | For Makers | Products | Community | Documentation | SHOP

UNO Family
Shields

Arduino Motor Shield Rev3	Arduino 4 Relays Shield	Arduino Ethernet Shield Rev2	Arduino 9 Axis Motion Shield
Arduino UNO SPE Shield			

Embedded Systems and IOT

For Professionals | For Education | For Makers | Products | Community | Documentation | SHOP

Classic Family

In the Classic Family, you will find boards such as the Leonardo, Micro & Zero.

Boards

Arduino Leonardo	Arduino Micro	Arduino Zero

Embedded Systems and IOT

For Professionals | For Education | For Makers | Products | Community | Documentation | SHOP

Mega Family

In the Mega family, you will find the boards for the projects that demand lots of computing power and GPIO pins.

Arduino Mega 2560 Rev3	Arduino Due	Arduino GIGA R1 WiFi	Arduino GIGA Display Shield

Embedded Systems and IOT

For Professionals | For Education | For Makers | Products | Community | Documentation | SHOP

Modulino

Arduino Modulino: connect to your board via Qwiic and bring plug-and-play sensors, actuators & more to your projects.

Modulino Buttons	Modulino Buzzer	Modulino Distance	Modulino Knob
Modulino Movement	Modulino Pixels	Modulino Thermo	

Embedded Systems and IOT

History

Table 2-1. Timeline of Arduino products

Board name	Year	Microcontroller	Board name	Year	Microcontroller
Diecimila	2007	ATmega168V	Mega 2560	2010	ATmega2560
LilyPad	2007	ATmega168V/ATmega328V	Uno	2010	ATmega328P
Nano	2008	ATmega328/ATmega168	Ethernet	2011	ATmega328
Mini	2008	ATmega168	Mega ADK	2011	ATmega2560
Mini Pro	2008	ATmega328	Leonardo	2012	ATmega32U4
Duemilanove	2008	ATmega168/ATmega328	Esplora	2012	ATmega32U4
Mega	2009	ATmega1280	Micro	2012	ATmega32U4
Fio	2010	ATmega328P	YÚN	2013	ATmega32U4 + Linino

Embedded Systems and IOT

Arduino-Specific AVR Microcontrollers

Table 2-8. AVR microcontrollers used in Arduino products

Microcontroller	Flash	I/O pins (max)	Notes
ATmega168	16K	23	20 MHz clock
ATmega168V	16K	23	10 MHz clock
ATmega328	32K	23	20 MHz clock
ATmega328P	32K	23	20 MHz clock, picoPower
ATmega328V	32K	23	10 MHz clock
ATmega1280	128K	86	16 MHz clock
ATmega2560	256K	86	16 MHz clock
ATmega32U4	32K	26	16 MHz clock

Embedded Systems and IOT

Atmega 168/328

ATmega168/328 microcontroller internal block diagram

Embedded Systems and IOT

Atmega 168/328 characteristics

Memory

The ATmega328 has twice the amount of each type of memory as the ATmega168, as shown in Table 2-9. Other than this, the two parts are identical.

Table 2-9. ATmega168/328 on-board memory

	ATmega168	ATmega328
Flash program memory	16K bytes	32K bytes
EEPROM	512 bytes	1K bytes
RAM	1K bytes	2K bytes

Embedded Systems and IOT

Atmega 168/328 Features

These two parts share the following features:

- In-system programming by on-chip boot program
- Two 8-bit timer/counters with separate prescaler and compare mode
- One 16-bit timer/counter with separate prescaler, compare mode, and capture mode
- Real-time counter with separate oscillator
- Six PWM channels
- Six- or eight-channel 10-bit ADC (depends on package type)
- USART
- Master/slave SPI serial interface
- Two-wire serial interface (Philips I2C compatible)
- Programmable watchdog timer
- Analog comparator
- 23 programmable I/O lines

Embedded Systems and IOT

Atmel ATmega328/P

Packages

The ATmega168 and ATmega384 are available in four different package types:

28-pin DIP, 28-pin MLF surface-mount, 32-pin TQFP surface-mount, and 32-pin MLF surface-mount.

The 28-pin DIP is the most commonly used package on Arduino boards, although the Uno SMD uses a 32-pin surface-mount package. This section will focus on the PDIP version of the ATmega168.

Pin Configurations

Pin-out

Figure 5-1. 28-pin PDIP

Embedded Systems and IOT

Atmega 168/328 Features

Pin Configurations

Pin-out

Figure 5-1. 28-pin PDIP

Embedded Systems and IOT

Atmega 168/328 Features

Ports

The ATmega168/328 has three ports designated as B, C, and D.

- Ports B and D are 8- bit ports.
- Port C has six pins available that can be used as ADC inputs.
  - PC4 and PC5 are also connected to the TWI logic and provide SCL and SDA I2C-compatible signals (clock and data).
  - Also note that PC6 is typically used as the RESET input.
  - There is no PC7 on the ATmega168/328 parts. Also note that there is no Port A in the ATmega168/328 part. Each port provides bidirectional discrete digital I/O with programmable internal pull-up resistors.

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Aim: Implement Seven Segment Display in Arduino Using Proteus

Explanation:
The 7-segment displays are really just seven LEDs lined up in a particular pattern.

Procedure:

1. **Create circuit**
2. **Copy the code and execute it on Arduino IDE.**
3. **Upload the code.**
4. **Run the project.**

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Output:

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Output:

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Output:

```
int a=13;
int b=12;
int c=11;
int d=10;
int e=9;
int f=8;
int g=7;

void setup()
{
pinMode(a,OUTPUT);
pinMode(b,OUTPUT);
pinMode(c,OUTPUT);
pinMode(d,OUTPUT);
pinMode(e,OUTPUT);
pinMode(f,OUTPUT);
pinMode(g,OUTPUT);
}
```

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Code cont. :

```
void loop() {
//for 0:
digitalWrite (a,1);
digitalWrite (b,1);
digitalWrite (c,1);
digitalWrite (d,1);
digitalWrite (e,1);
digitalWrite (f,1);
digitalWrite (g,0);
delay (600);

//for 1:
digitalWrite (a,0);
digitalWrite (b,1);
digitalWrite (c,1);
digitalWrite (d,0);
digitalWrite (e,0);
digitalWrite (f,0);
digitalWrite (g,0);
delay (600);
```

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Code cont. :

```
//for 2:
digitalWrite (a,1);
digitalWrite (b,1);
digitalWrite (c,0);
digitalWrite (d,1);
digitalWrite (e,1);
digitalWrite (f,0);
digitalWrite (g,1);
delay (600);

//for 3 CS:
digitalWrite (a,1);
digitalWrite (b,1);
digitalWrite (c,1);
digitalWrite (d,1);
digitalWrite (e,0);
digitalWrite (f,0);
digitalWrite (g,1);
delay (600);
```

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Code cont. :

```
//for 4:
digitalWrite (a,0);
digitalWrite (b,1);
digitalWrite (c,1);
digitalWrite (d,0);
digitalWrite (e,0);
digitalWrite (f,1);
digitalWrite (g,1);
delay (600);

//for 5:
digitalWrite (a,1);
digitalWrite (b,0);
digitalWrite (c,1);
digitalWrite (d,1);
digitalWrite (e,0);
digitalWrite (f,1);
digitalWrite (g,1);
delay (600);
```

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Code cont. :

```
//for 6:
digitalWrite (a,1);
digitalWrite (b,0);
digitalWrite (c,1);
digitalWrite (d,1);
digitalWrite (e,1);
digitalWrite (f,1);
digitalWrite (g,1);
delay (600);

//for 7
digitalWrite (a,1);
digitalWrite (b,1);
digitalWrite (c,1);
digitalWrite (d,0);
digitalWrite (e,0);
digitalWrite (f,0);
digitalWrite (g,0);
delay (600);
```

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Code cont. :

```
//for 8:
digitalWrite (a,1);
digitalWrite (b,1);
digitalWrite (c,1);
digitalWrite (d,1);
digitalWrite (e,1);
digitalWrite (f,1);
digitalWrite (g,1);
delay (600);

//for 9:
digitalWrite (a,1);
digitalWrite (b,1);
digitalWrite (c,1);
digitalWrite (d,1);
digitalWrite (e,0);
digitalWrite (f,1);
digitalWrite (g,1);
delay (600);
}
```

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Output:

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Output:

Embedded Systems and IOT

Project 2

Seven Segment Display in Arduino

Output:

Embedded Systems and IOT

Quiz

Write the C code required for implementing the following circuit;

The circuit consists of a Seven Segment Display connected to an Arduino board. The display will show the counts from 0 to 9 after each click on a push button. Additionally, an indicator LED will light up to indicate each press on the button.

Questions

THANK YOU