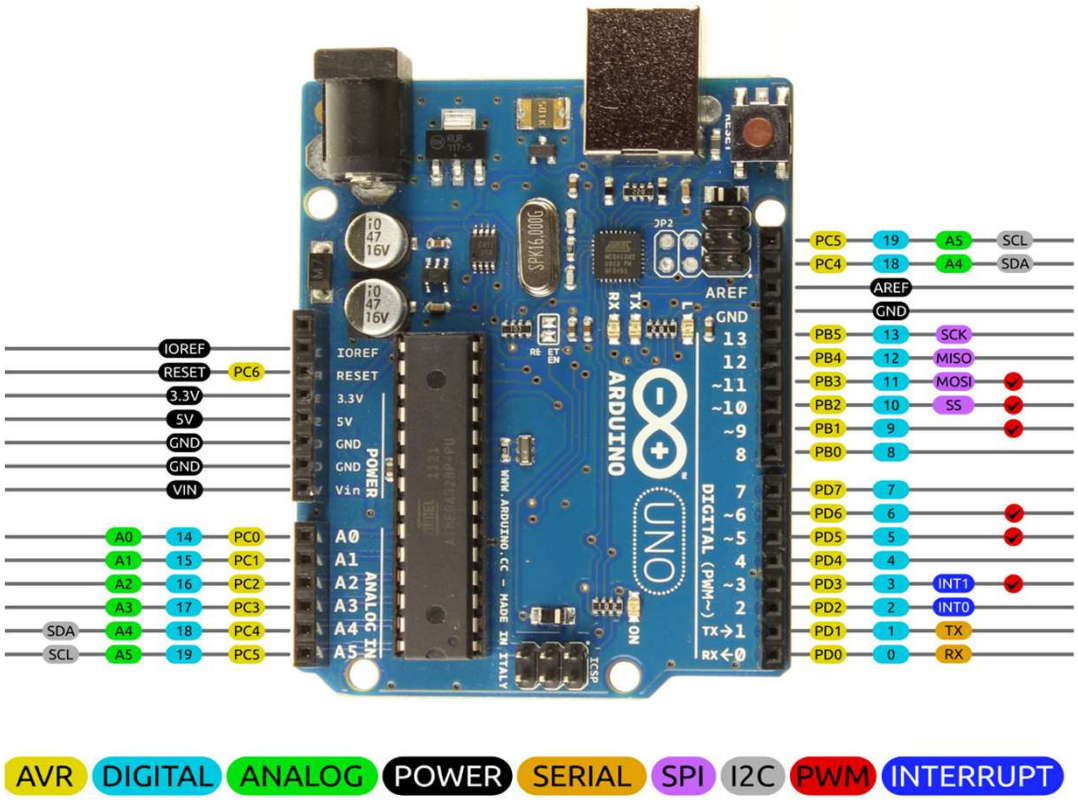
EE3704 Embedded System

Chapter 7

Presented by
Asst. Prof. Dr.Narong Aphiratsakun

Chapter 7: Timer and External Interrupts

Arduino Uno R3 Pinout



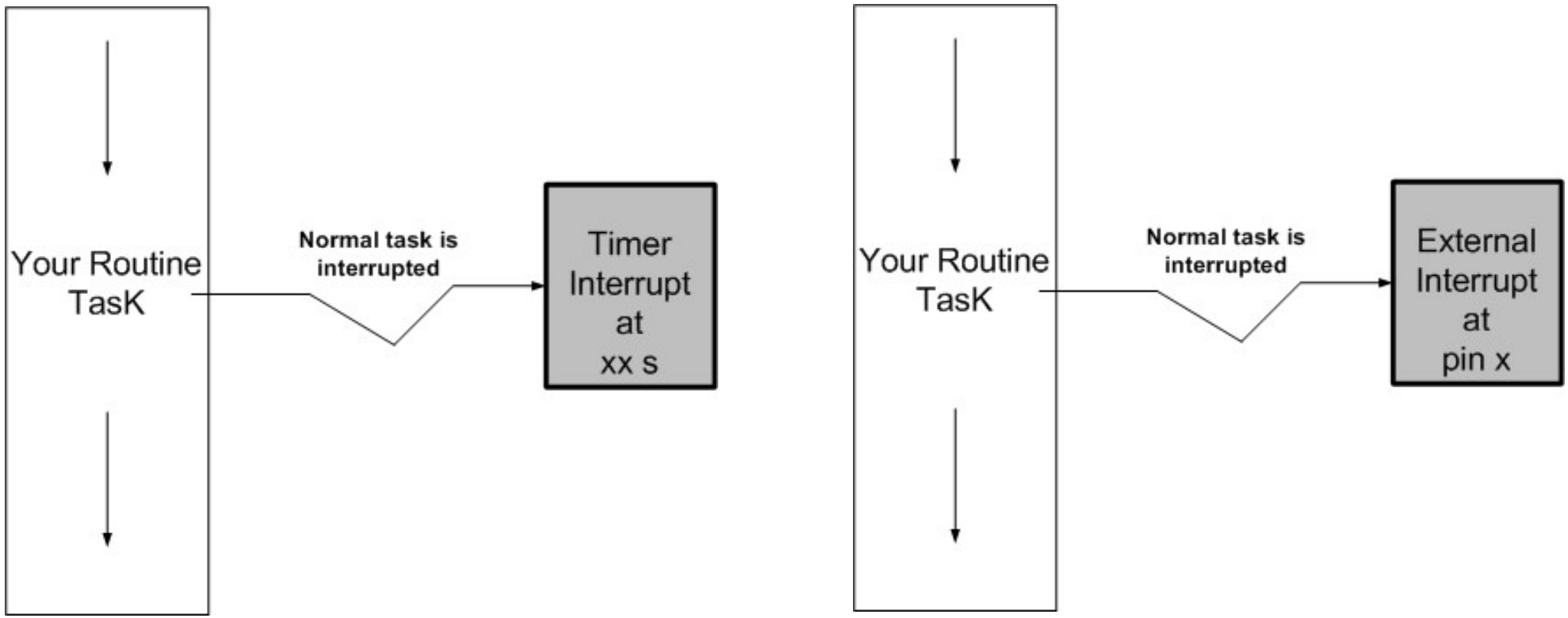
AVR | DIGITAL | ANALOG | POWER | SERIAL | SPI | I2C | PWM | INTERRUPT

Function:

```
noInterrupts();    //disable
interrupts();      //enable

//Timer
ISR(TIMER1_COMPA_vect)
//External
attachInterrupt();
```

Chapter 7 : Timer and External Interrupts



Chapter 7 : Timer and External Interrupts

PWM and timer

There is fixed relation between the timers and the PWM capable outputs. When you look in the data sheet or the pinout of the processor these PWM capable pins have names like OCRxA, OCRxB or OCRxC (where x means the timer number 0..5). The PWM functionality is often shared with other pin functionality.

The Arduino has 3Timers and 6 PWM output pins. The relation between timers and PWM outputs is:
Pins 5 and 6: controlled by timer0
Pins 9 and 10: controlled by timer1
Pins 11 and 3: controlled by timer2

On the Arduino Mega we have 6 timers and 15 PWM outputs:
Pins 4 and 13: controlled by timer0
Pins 11 and 12: controlled by timer1
Pins 9 and10: controlled by timer2
Pin 2, 3 and 5: controlled by timer 3
Pin 6, 7 and 8: controlled by timer 4
Pin 46, 45 and 44:: controlled by timer 5

Chapter 7 : Timer and External Interrupts

Timer0:
Timer0 is a 8bit timer.
In the Arduino world timer0 is been used for the timer functions, like delay() 312, millis() 720 and micros() 325 . If you change timer0 registers, this may influence the Arduino timer function. So you should know what you are doing.

Timer1:
Timer1 is a 16bit timer.
In the Arduino world the Servo library 541 uses timer1 on Arduino Uno (timer5 on Arduino Mega).

Timer2:
Timer2 is a 8bit timer like timer0.
In the Arduino work the tone() 406 function uses timer2.

Timer3, Timer4, Timer5:
Timer 3,4,5 are only available on Arduino Mega boards. These timers are all 16bit timers.

Chapter 7 : Timer and External Interrupts

Clock select and timer frequency

Different clock sources can be selected for each timer independently. To calculate the timer frequency (for example 2Hz using timer1) you will need:

1. CPU frequency 16Mhz for Arduino
2. maximum timer counter value (256 for 8bit, 65536 for 16bit timer)
3. Divide CPU frequency through the choosen prescaler (16000000 / 256 = 62500)
4. Divide result through the desired frequency (62500 / 2Hz = 31250)
5. Verify the result against the maximum timer counter value (31250 < 65536 success) if fail, choose bigger prescaler.

Table 16-5. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Table 13-4. Waveform Generation Mode Bit Description[1]

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Chapter 7 : Timer and External Interrupts

Example Timer 1 (A is normally use for triggering and B is use for comparing)

- TCCR1A = 0; // set entire TCCR1A and B register to 0
- TCCR1B = 0;
- TCNT1 = 0; //initialize counter value to 0

- TCCR1B |= (1 << WGM12); // turn on CTC (clear timer on compare) mode
- TCCR1B |= (1 << CS12); // Set prescaler
- TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt

OCR1A // value for set up timer (the compare match value)

Chapter 7 : Timer and External Interrupts

Example 7.1: Interrupt at 0.5s

- ON buzzer a beep 0.1s for every 0.5s

T = 0.5s, → f = 2Hz

16M/256= 62500

Therefore 62500/2 = 31250

Timer register

prescaler

Chapter7_Example_1

```
int LedPin =  13;
void setup()
{
  pinMode(LedPin, OUTPUT);
  // initialize timer1
  noInterrupts();    // disable all interrupts
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1  = 0;

  OCR1A = 31250;            // compare match register 16MHz/256/2Hz
  TCCR1B |= (1 << WGM12);   // CTC mode
  TCCR1B |= (1 << CS12);    // 256 prescaler
  TIMSK1 |= (1 << OCIE1A);  // enable timer compare interrupt
  interrupts();             // enable all interrupts
}

ISR(TIMER1_COMPA_vect)          // timer compare interrupt service routine
{
  Do your task
}

void loop()
{
  // your program here...
}
```

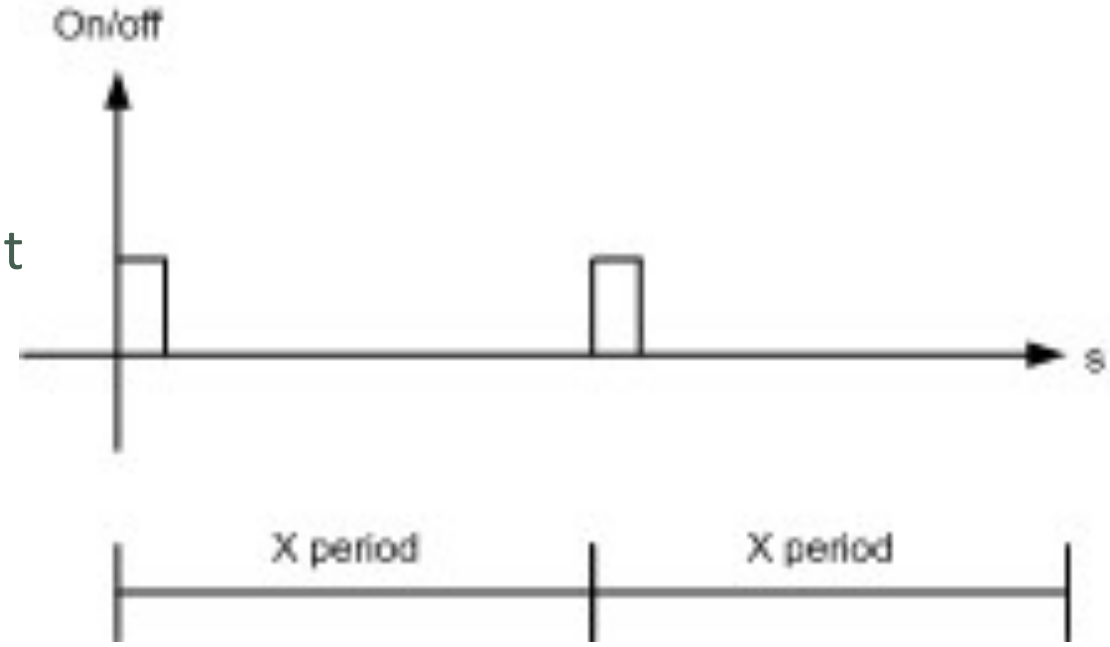
- Show circuit diagram, Coding and result

Chapter 7 : Timer and External Interrupts

Example 7.1: Interrupt at 0.5s

- ON buzzer a beep 0.1s for every 0.5s

- Show circuit diagram, Coding, result



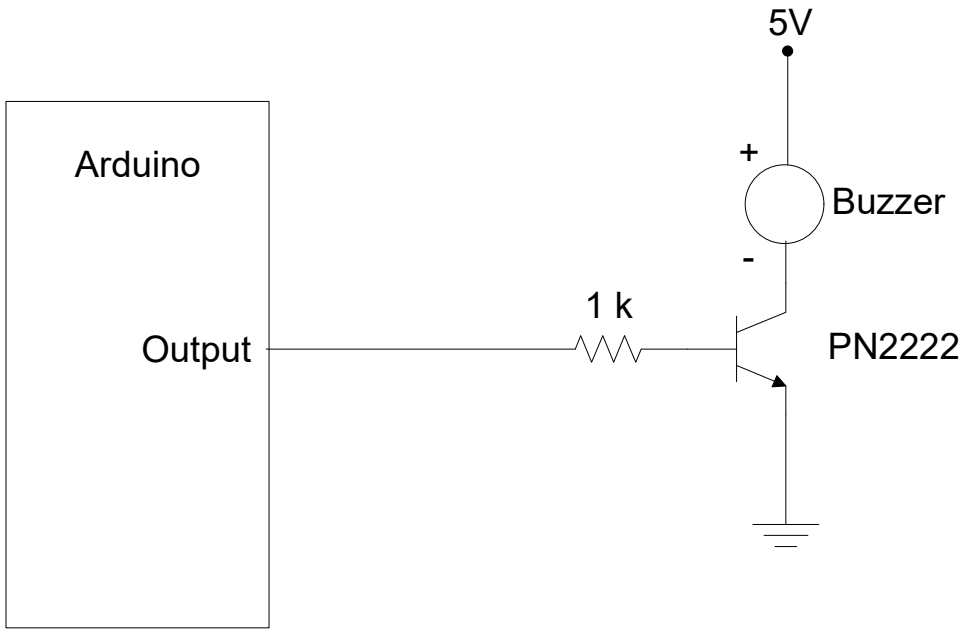
$f = 0.25s \quad T = 4s$

$16M/256 = 62500$ then $\frac{62500}{0.25} = 250000 > 65535$

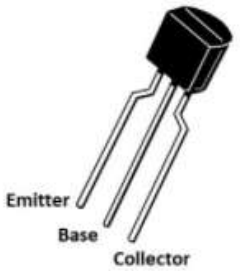
Chapter 7 : Timer and External Interrupts

Example 7.1: Interrupt at 0.5s

- ON buzzer a beep 0.1s for every 0.5s



PN2222



Chapter 7 : Timer and External Interrupts

Example 7.2: Interrupt at 4s

- ON buzzer a beep 0.1s for every 4s

$T = 4\ s \rightarrow f = 0.25\ Hz$

$16M / 1024 = 15625$

$\frac{15625}{0.25} = 62500$

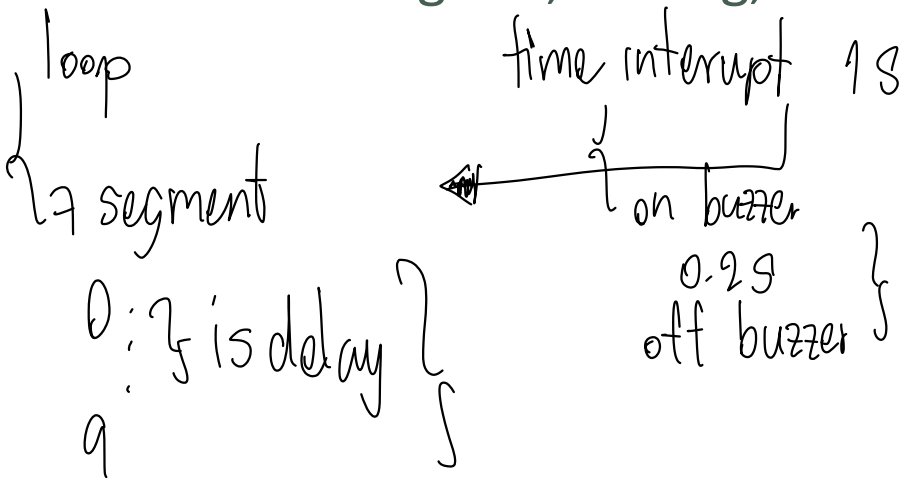
- Show circuit diagram, Coding, result
  - Tinkercad
  - Real Arduino board

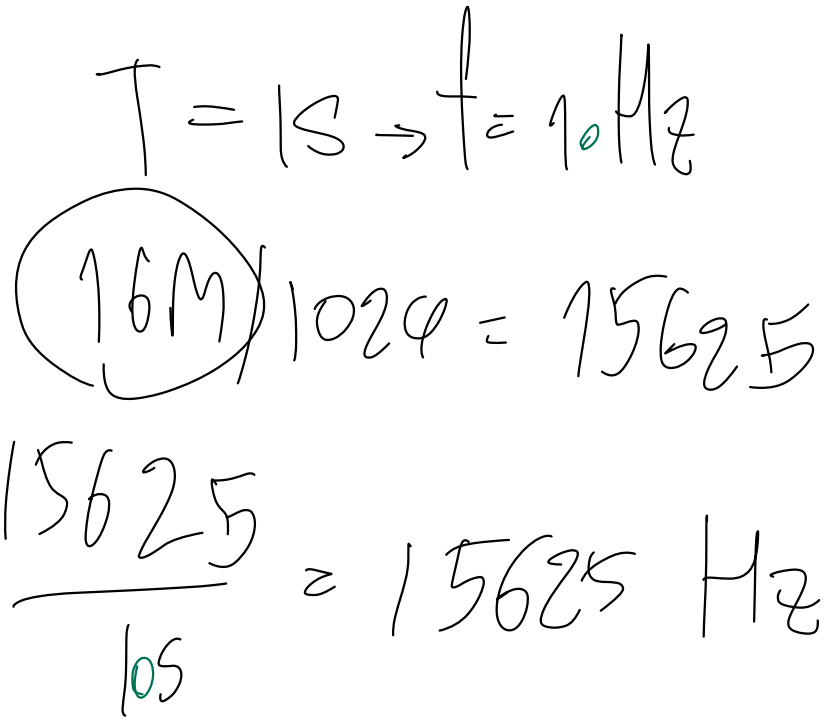
Chapter 7 : Timer and External Interrupts

Example 7.3

- Running (CC/CA) 7 segment 0 - 9 delay 1 second
- Timer interrupt every 1 second, switch ON Buzzer Pin4 for 0.2s, and switch OFF for 0.8s.

- Show circuit diagram, Coding, result

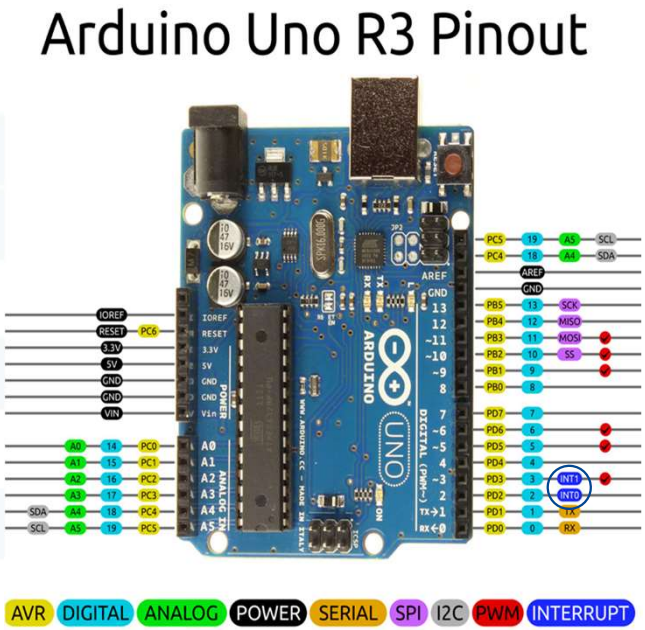


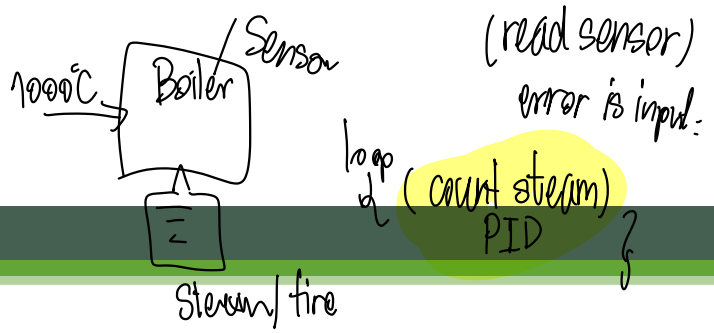


Chapter 7 : Timer and External Interrupts

- External Interrupt

BOARD	DIGITAL PINS USABLE FOR INTERRUPTS
Uno, Nano, Mini, other 328-based	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR1000 Rev.1	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Due	all digital pins
101	all digital pins (Only pins 2, 5, 7, 8, 10, 11, 12, 13 work with CHANGE)





Chapter 7 : Timer and External Interrupts

- External Interrupt

attachInterrupt()

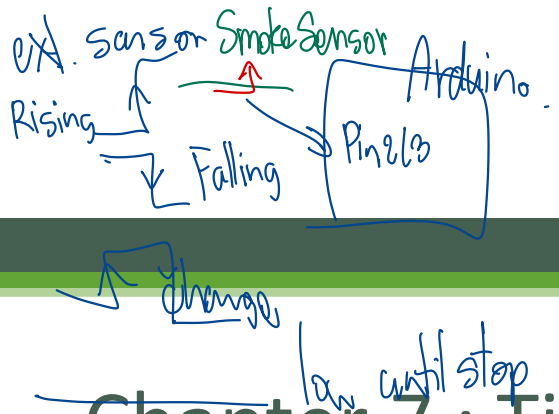
[External Interrupts]

Description

Digital Pins With Interrupts

The first parameter to `attachInterrupt()` is an interrupt number. Normally you should use `digitalPinToInterrupt(pin)` to translate the actual digital pin to the specific interrupt number. For example, if you connect to pin 3, use `digitalPinToInterrupt(3)` as the first parameter to `attachInterrupt()`.

BOARD	DIGITAL PINS USABLE FOR INTERRUPTS
Uno, Nano, Mini, other 328-based	2, 3
Uno WiFi Rev.2	all digital pins
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7



Chapter 7 : Timer and External Interrupts

Syntax

`attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);` (recommended)
`attachInterrupt(interrupt, ISR, mode);` (not recommended)
`attachInterrupt(pin, ISR, mode);` (not recommended Arduino Due, Zero, MKR1000, 101 only)

Parameters

`interrupt`: the number of the interrupt (`int`)
`pin`: the pin number *(Arduino Due, Zero, MKR1000 only)*
`ISR`: the ISR to call when the interrupt occurs; this function must take no parameters and return nothing. This function is sometimes referred to as an interrupt service routine.
`mode`: defines when the interrupt should be triggered. Four constants are predefined as valid values:

- LOW to trigger the interrupt whenever the pin is low,
- CHANGE to trigger the interrupt whenever the pin changes value
- RISING to trigger when the pin goes from low to high,
- FALLING for when the pin goes from high to low.

Chapter 7 : Timer and External Interrupts

Example 7.4

- Running (CC/CA) 7 segment 0 - 9 with delay 1 second
- If interrupt SW is pressed
  - Buzzer On for a short beep (0.2s)

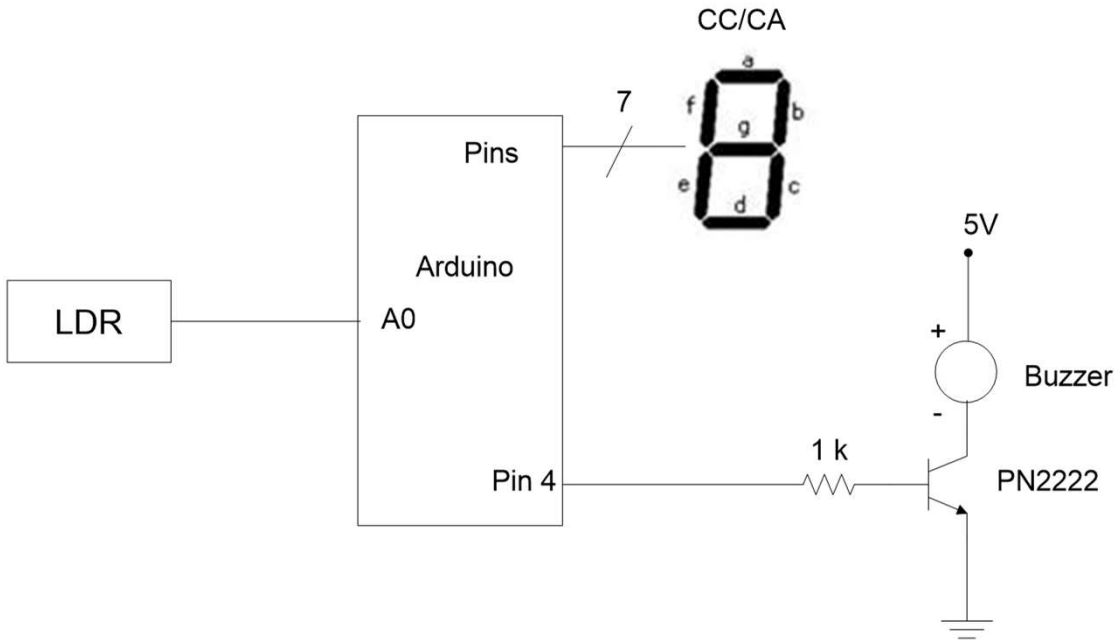
- Show circuit diagram, Coding, result

Chapter 7 : Timer and External Interrupts

Example 7.5 (for strong one)

- Running (CC/CA) 7 segment 0 - 9 with delay 1 second
- If LDR is dark (dark as an interrupt case)
  - Buzzer On for a short beep (0.2s)

- Show circuit diagram, Coding, result




ASSUMPTION UNIVERSITY OF THAILAND
SINCE 1969
มหาวิทยาลัยอัสสัมชัญ
OMNIA
VINCIT