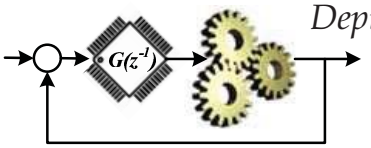
Chapter 7
ATmega328 Interfacing with High Voltage / Current

Assoc. Professor Dr. Rosbi bin Mamat

rosbi@fke.utm.my



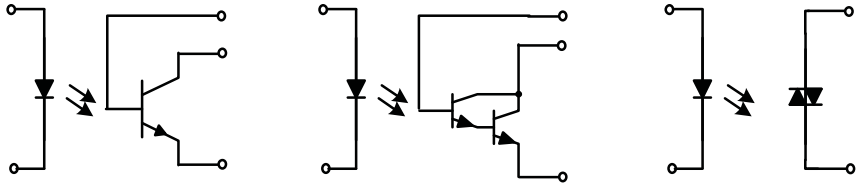
Dept. of Control & Mechatronics Engineering
Faculty of Electrical Engineering
Universiti Teknologi Malaysia

7.1 Isolation

- When high current or high voltage devices are connected to a μC system, **Isolation** must be made between high voltage/current devices & μC system to avoid high voltage/current damages μC or human!.
- **Optoisolator** / **Optocoupler**
  - Optoisolator uses LED (as transmitter) & opto detector (as receiver) to send signal through light. LED is connected to output port of μC & photo-transistor output is connected to high voltage/current device.
  - $\because$ LED & opto detector are separated by insulated air, isolation between μC & devices is achieved.

7.2 Isolation – Optoisolator

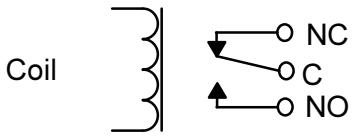
- **Optoisolator** / **Optocoupler**
  - Optoisolator is useful to control devices which are slightly remote from μC system & devices which have separate power supply from μC.
  - Sample types of optoisolator:



Transistor Output (4N26) | Darlington Output (MOC119) | Triac Output (MOC3011)

7.3 Isolation – Electromagnetic Relay

- **Electromagnetic Relay** can be used to control high voltage/current devices.
  Structure of a relay:



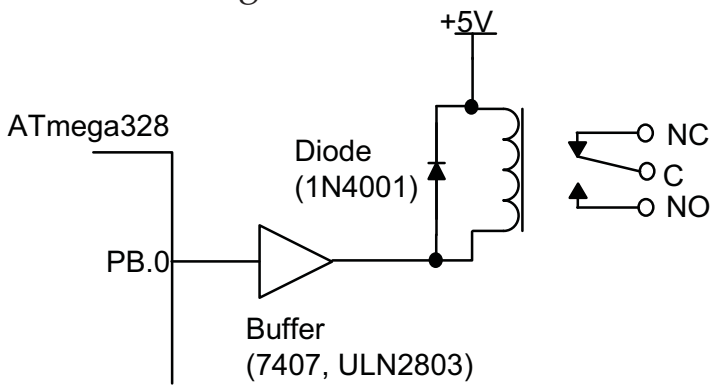
C - common terminal
NC - normally closed terminal
NO - normally opened terminal

  - Relay can control devices with DC or AC voltages. Must choose suitable relay with specific devices.

7.3 Isolation – Electromagnetic Relay

Electromagnetic Relay

- For relays with low coil voltage (<= 5 volts) & small current, the following circuit can be used:

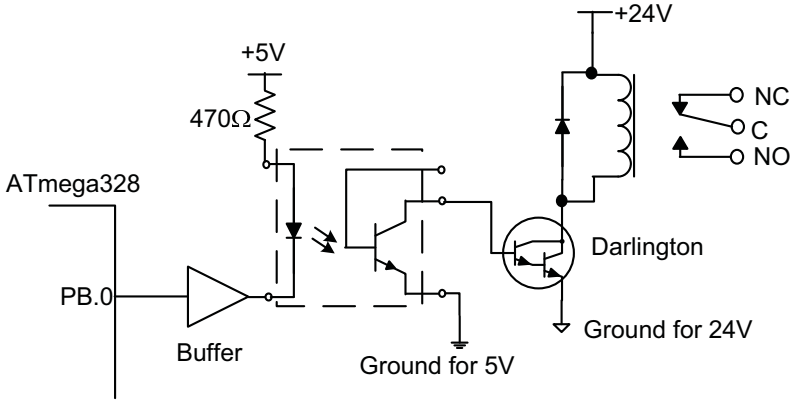


- The diode is used to the avoid negative voltage induced in the coil from damaging the buffer & ATmega328.

7.3 Isolation – Electromagnetic Relay

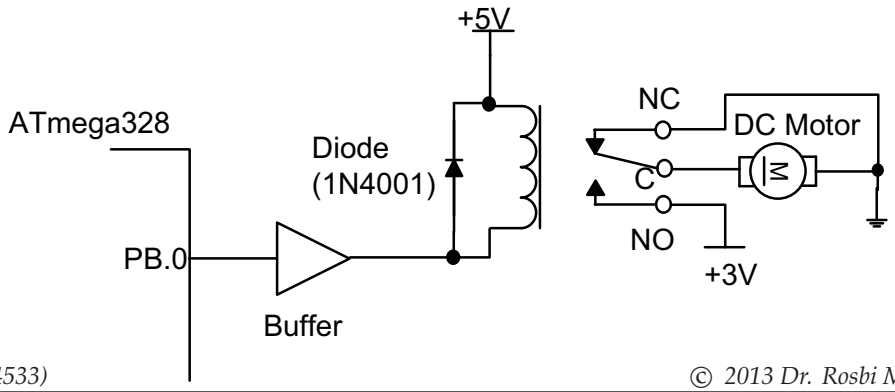
Electromagnetic Relay

- When a relay with high voltage (> 5 volts) & high current coil, or when the relay output need to be connected to AC devices, an optoisolator can be used with a relay. For example, for a relay with +24V coil voltage, the following circuit can be used:



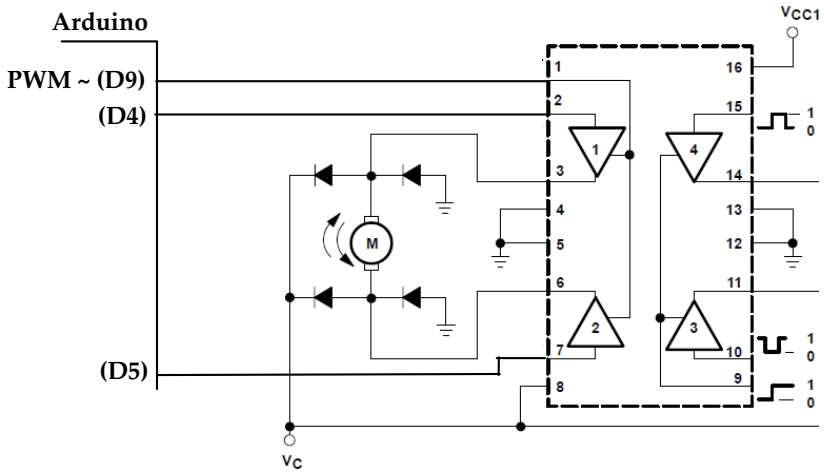
7.4 Interfacing With DC Motor

- **Example:** *Interfacing a small DC motor (+3V, 500mA) to ATmega328.*
- **Solution:** The DC motor **SHOULD NOT** be connected directly to ATmega328 because the 500mA current can burn the ATmega328. We can use +5V relay. When `PB0='1'`, motor stops & When `PB0='0'`, motor rotates (one way only!).



7.4 Interfacing With DC Motor

- **Example:** *Interfacing a small DC motor (+3V, 500mA) to ATmega328.*
- **Solution:** Use a motor driver IC such as L293. Direction of rotation is set by D4 & D5. Motor speed is controlled by Pulse Width Modulation (PWM) on pin D9.



7.4 Interfacing With DC Motor

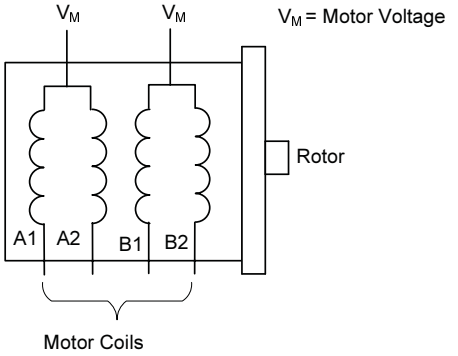
- **Example:** *Interfacing a small DC motor (+3V, 500mA) to ATmega328.*
- **Solution:** Sample program in Arduino:

```
void setup() {
  pinMode(D4, OUTPUT);  // D4 & D5 for motor
  pinMode(D5, OUTPUT);  // direction control
  analogWrite(9, 0);    // stop motor
}

void loop() {
  digitalWrite(D4, HIGH); // rotate clockwise
  digitalWrite(D5, LOW);
  analogWrite(9, 255);    // full speed
  delay(10000);           // wait 10 secs
  analogWrite(9, 0);      // stop motor
}
```


7.5 Interfacing with Stepper Motor

- **Stepper motor** – a type of motor which rotates step-by-step in a specific angle.
- Common angles for each step are: 15^o & 7.5^o. For a stepper motor with step angle 15^o, a complete rotation (360^o) requires 24 steps.
- Stepper motors are used in printers, disc drives & robots.



7.5 Interfacing with Stepper Motor

- Comparison between stepper motor & DC motor:

Stepper Motor	DC Motor
Rotates in steps	Rotates continuosly
Can be controlled in open-loop. Does not require sensor for position feedback	Must be controlled in closed-loop. Requires position sensor for feedback
Control software is easier to write	Complex control software

- Driving a stepper motor involves applying voltages to motor coils (A1, A2, B1 & B2) in a specific sequence.

7.5 Interfacing with Stepper Motor

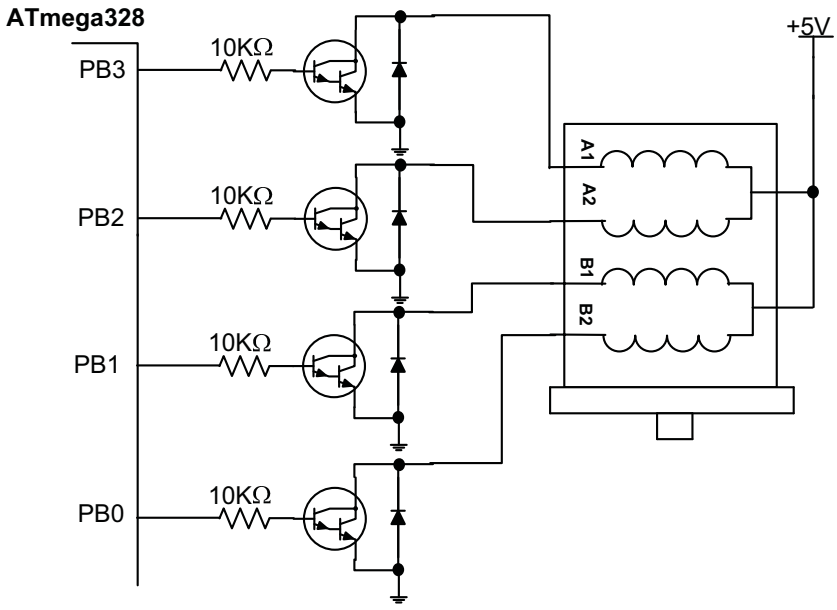
- In full-step mode, the following sequence of voltages must be applied to motor coils.

Sequence	A1	A2	B1	B2	Code (Hex)
1	1	0	1	0	$0A
2	1	0	0	1	$09
3	0	1	0	1	$05
4	0	1	1	0	$04

- '1' on A1, A2, B1 & B2 means V_M Volts is applied. '0' on A1, A2, B1 & B2 means 0V is applied.

7.5 Interfacing with Stepper Motor

- Direct drive circuit: Example of connecting stepper motor AIRPAX 5V with 7.5^o per step.



7.5 Interfacing with Stepper Motor

- To step the motor in clock-wise direction this sequence of codes : \$0A, \$09, \$05, \$06 are repeatedly sent to Port B (lower nibble only).
- To step the motor in counter clock-wise direction this sequence of codes: \$06, \$05, \$09, \$0A are repeatedly sent to Port B.
- Normally, a delay is performed between the codes sent to allow the motor to reach steady-state & to produce rotation in a specified speed.

7.5 Interfacing with Stepper Motor

- Stepper motor driver chips e.g. SAA1027, L297/298, UCN4204B etc. are available. Advantages of using these driver chips are:
  - Simplified the interfacing circuit with μC.
  - Simplified software writting since the programmer is not required to remember the sequence of codes sent.
  - Driver chips already have the required voltage/current capability to drive stepper motor directly.

7.5 Interfacing with Stepper Motor

- **Example:** *Write a program to rotate an AIRPAX stepper motor in previous slide for 60^o clock-wise & then 15^o counter clock-wise. This motor require 10ms to reach steady-state.*
- **Solution:** To rotate 60^o 8 steps are required & 15^o requires 2 steps.

7.5 Interfacing with Stepper Motor

- **Program:** Stepper motor control

```
void setup() {
  DDRB = 0x0F;  // PB0-PB3 as output
}

void loop() {
  int i;

  for (i = 0; i < 2; i++) {  // cw 60 degree
    PORTB = 0x0A; delay_ms(10);
    PORTB = 0x09; delay_ms(10);
    PORTB = 0x05; delay_ms(10);
    PORTB = 0x06; delay_ms(10);
  }
  // counter cw 15 degree
  PORTB = 0x05;  delay_ms(10);
  PORTB = 0x09;  delay_ms(10);
}
```


7.5 Interfacing with Stepper Motor

- **Program:** Stepper motor control – using look-up table.

```
const unsigned char cw60[]= {0x0A, 0x09, 0x05, 0x06,
                             0x0A, 0x09, 0x05, 0x06};
const unsigned char ccw15[]= {0x05, 0x09};

void setup() {
  DDRB = 0x0F;                // PB0-PB3 as output
}

void loop() {
  int i;

  for (i = 0; i < 8; i++) {   // clock-wise 60 degree
    PORTB = cw60[i];
    delay_ms(10);
  }

  for (i = 0; i < 2; i++) {  // counter cw 15 degree
    PORTB = ccw15[i];
    delay_ms(10);
  }
}
```