Attach Interrupt (interrupt, function, mode)

Description

Specifies a function to call when an external interrupt occurs. Replaces any previous function that was attached to the interrupt. Most Arduino boards have two external interrupts: numbers 0 (on digital pin 2) and 1 (on digital pin 3). The Arduino Mega has an additional four: numbers 2 (pin 21), 3 (pin 20), 4 (pin 19), and 5 (pin 18).

Parameters

Interrupt: the number of the interrupt (int) function: the function 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 contstants 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.

Using Interrupts

Interrupts are useful for making things happen automatically in microcontroller programs, and can help solve timing problems. A good task for using an interrupt might be reading a rotary encoder, monitoring user input. If you wanted to insure that a program always caught the pulses from a rotary encoder, never missing a pulse, it would make it very tricky to write a program to do anything else, because the program would need to constantly poll the sensor lines for the encoder, in order to catch pulses when they occurred. Other sensors have a similar interface dynamic too, such as trying to read a sound sensor that is trying to catch a click, or an infrared slot sensor (photo-interrupter) trying to catch a coin drop. In all of these situations, using an interrupt can free the microcontroller to get some other work done while not missing the doorbell.

Example

```
int pin = 13;
volatile int state = LOW;
void setup(){
 pinMode(pin, OUTPUT);
 attachInterrupt(0, blink, CHANGE);}
void loop(){
 digitalWrite(pin, state);}
void blink(){
 state = !state;}
```

Task 01

Construct a knight rider circuit with 8 LEDS and connect a separate LED to another pin called LED A.

Connect the switch with a de-bouncing circuit to the interrupt pin with a pull-up or down resistor.

The circuit should work as normal knight rider pattern lights should move when no switch input.

When you press the switch LED A should blink 5 times and continue your knight rider LED pattern from the LED that turns on before your switch input.

Give software and hardware solution for de-bouncing

Repeat the code for all the interrupt methods.

Submit the report pdf naming your group number with all the codes and screenshots.