Embedded System on AVR Microcontroller (ATMEGA32)

Exp3: Accepting Digital Inputs through PUSH Button Pull Up Network (Internal Pull Up/ External Pull Up) Pull Down Network (External Pull Up)

Submitted by
Ronit Dutta, MS in IOT and Signal Processing
Department of Electrical Engineering, IIT Kharagpur

Under the guidance of
Aurobinda Routray, Professor, Electrical Engineering

Department of Electrical Engineering
Indian Institute of Technology Kharagpur
January, 2024

- # Accept Digital Input on ATMEGA32

When we are trying to sense activity in the physical world using a microcontroller, the simplest activities we can sense are those in which we only need to know one thing about the physical world: Whether something is true or false. Is the switch is pressed or not? Are they touching the table or not? Is the door open or closed? In these cases, you can determine what you need to know using a digital input, or switch.

Digital or **binary** inputs to microcontrollers have two states: off and on. If current is flowing, the circuit is on. If it's not flowing, the circuit is off. To make a digital circuit, we need a circuit, and a movable conductor which can either complete the circuit, or not.

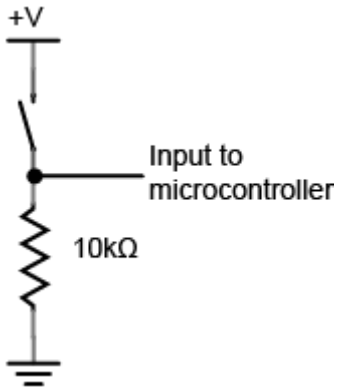


Fig: Schematic of a Digital Input to a Microcontroller

The above figure shows the electrical schematic for a digital input to a microcontroller. When the switch is closed, the current will flow through the resistance, and the microcontroller can then read the voltage. The microcontroller pin will then read as high voltage or HIGH. When the switch is open, the resistor connects the digital input to ground, so that it reads as zero voltage, or LOW.

- # What is Tactile switch?

A tactile switch is a switch whose operation is perceptible by press. These small sized switches are placed on PCBs and are used to close an electrical circuit when the button is pressed by a person. When the button is pressed, the switches turn ON and when the button is released, the switches turn OFF.

Most uses of Tactile Switches are to input an ON/OFF signal to devices.

- # Concept of Pull-up Network

Pull-up resistors are resistors used in logic circuits to ensure a well-defined logical level at a pin under all conditions. As a reminder, digital logic circuits have three logic states: high, low and floating (or high impedance). The high-impedance state occurs when the pin is not pulled to a high or low logic level, but is left “floating" instead. A good illustration of this is an unconnected input pin of a microcontroller. It is neither in a high or low logic state, and the microcontroller might unpredictably interpret the input value as either a logical high or logical low. Pull-up resistors are used to solve the dilemma for the microcontroller by pulling the value to a logical high state, as seen in the follow figure.

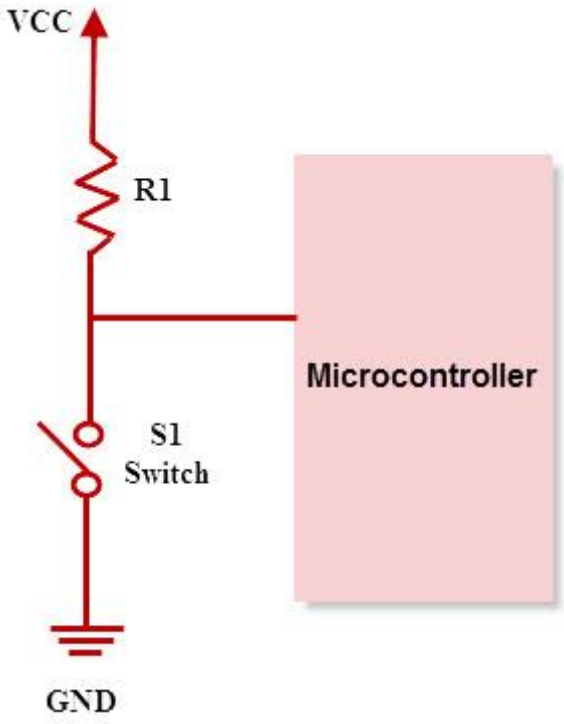


Fig: Pull-up Network

Without the pull-up resistor, the MCU’s input would be floating when the switch is open and pulled down to a logical low only when the switch is closed. Pull-up resistors are not a special kind of resistors; they are simply fixed-value resistors connected between the voltage supply (typically +5 V, +3.3 V, or +2.5 V) and the appropriate pin, which results in defining the input or output voltage in the absence of a driving signal. A typical pull-up resistor value is 10 kΩ, but can vary depending on the application.

- # Concept of Pull-down Network

Pull-down resistors work in the same manner as pull-up resistors, except that they pull the pin to a logical low value. They are connected between ground and the

appropriate pin on a device. An example of a pull-down resistor in a digital circuit can be seen in the following figure.

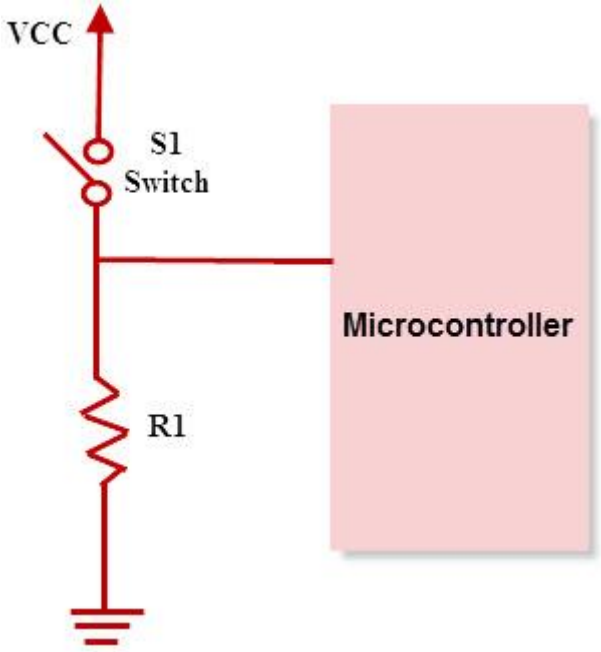


Fig: Pull-down Network

In the above figure a pushbutton switch is connected between the supply voltage and a microcontroller pin. In such a circuit, when the switch is closed, the microcontroller input is at a logical high value, but when the switch is open, the pull-down resistor pulls the input voltage down to ground (logical zero value), preventing an undefined state at the input.

- ## Steps to accept Digital Input on ATMEGA32

1. Write 0 in respective bit position in the DDR Register
2. Enable the internal Pull-up resistor (if needed)
3. Read the status of the pin using PIN register

- ## Accepting Digital Input through External Pull-up Network and show the status through LED

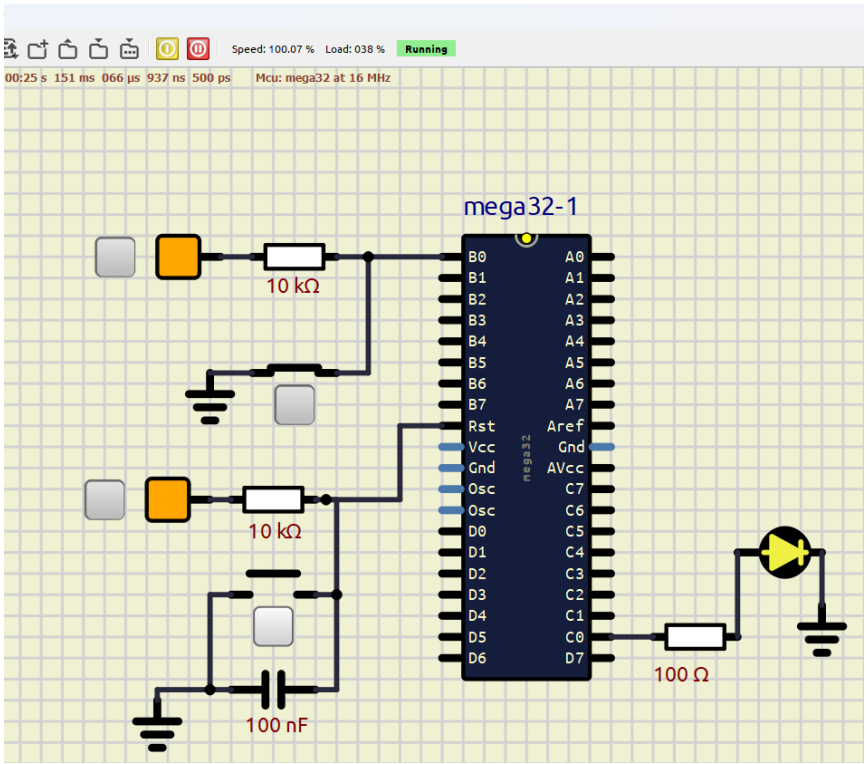
```
// Accepting Digital Input through External Pull-up Network
// Show the status through LED
// Pressed Tactile Switch: LED Glow
// Released Tactile Switch: LED Off

.INCLUDE "M32DEF.INC"
.ORG 0X0000
```

```
LDI R16,HIGH(RAMEND)
OUT SPH,R16
LDI R16,LOW(RAMEND)
OUT SPL,R16

SBI DDRC,PINC0 // To show the status through LED

// Make PORTB0 to work as input pin

CBI DDRB,PINB0

LDI R17,0x01
MAIN: CBI PORTC,PINC0
Loop: IN R16,PINB
        AND R16,R17
        BRNE MAIN
        SBI PORTC,PINC0
        JMP Loop
```

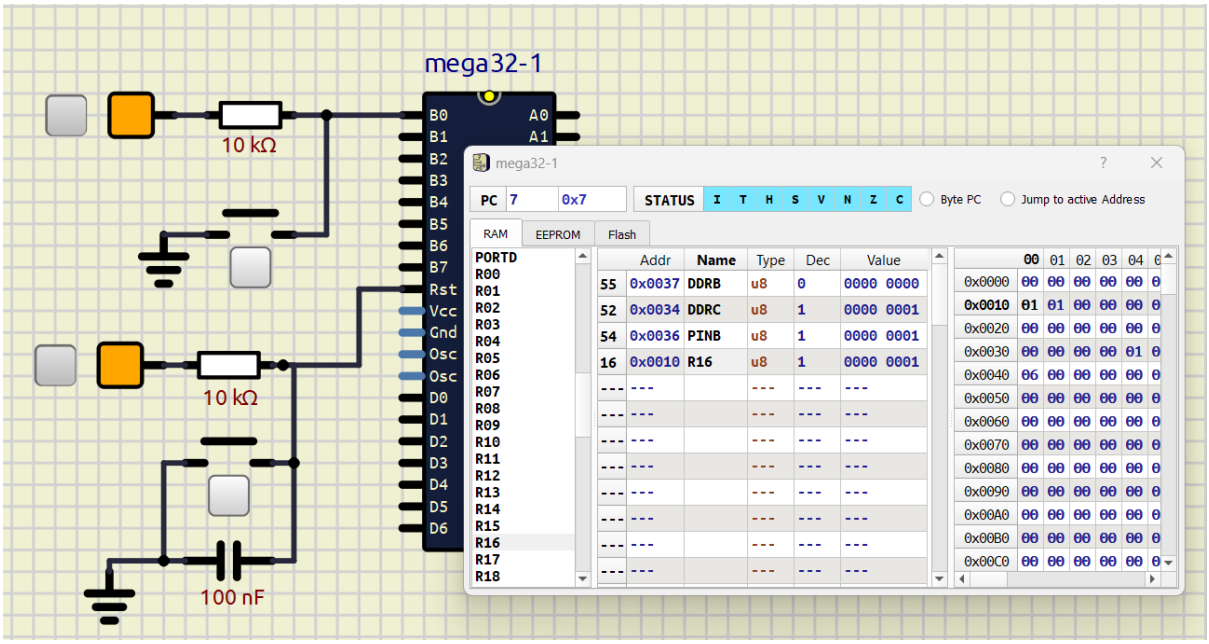
- **Make the below Circuit on SimulIDE to simulate**

Components Required:

1. ATmega32
2. Push Switches
3. 10K Ohm Resistors
4. 100 Ohm Resistor
5. Fixed Voltage
6. 100nF Capacitor



After uploading the HEX file, verify the simulation by TA and Show the MCU Monitor Status.



Assignment1: Write the C code for the above Experiment

- **Accepting Digital Input through Internal Pull-up Network and show the status through LED**

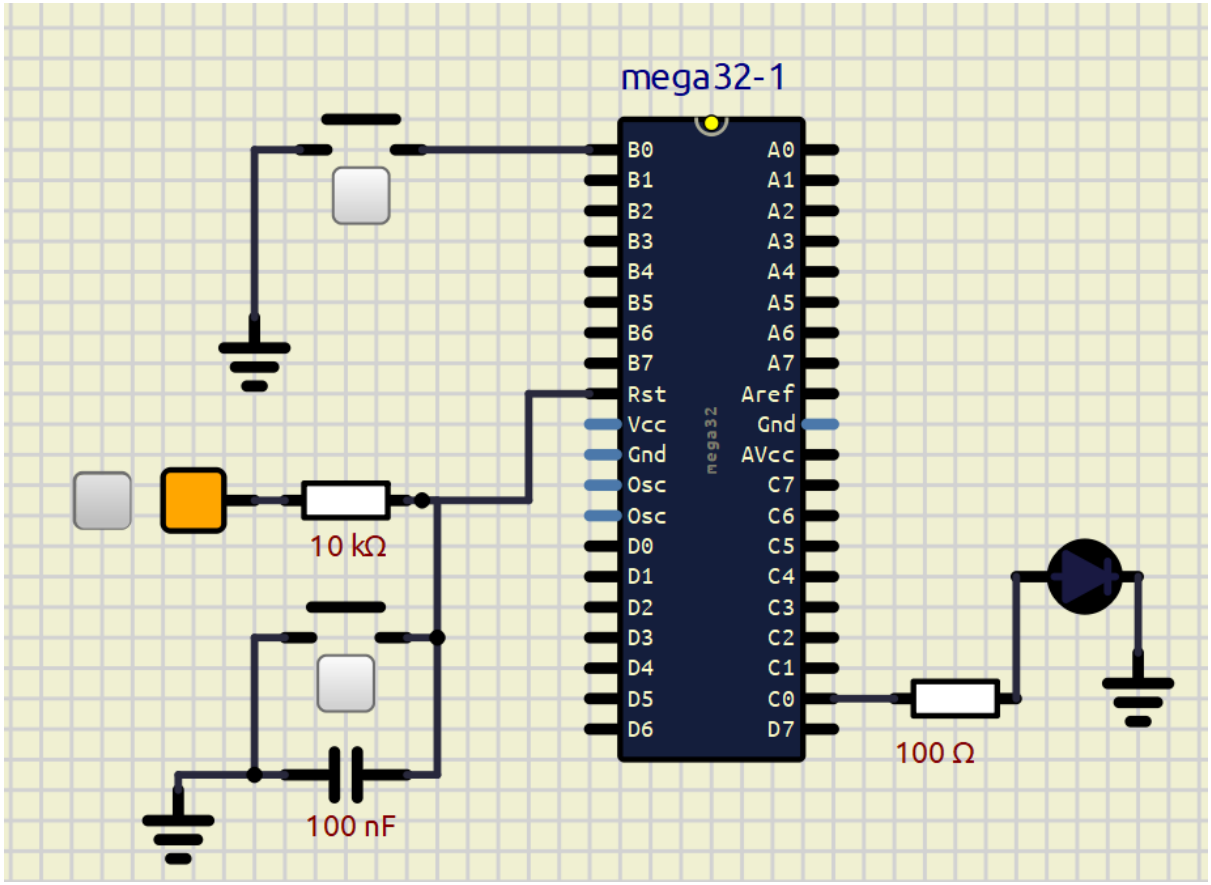
```
// Accepting Digital Input through Internal Pull-up Network
// Show the status through LED
// Pressed Tactile Switch: LED Glow
// Released Tactile Switch: LED Off

.INCLUDE "M32DEF.INC"
.ORG 0X0000

LDI R16,HIGH(RAMEND)
OUT SPH,R16
LDI R16,LOW(RAMEND)
OUT SPL,R16

SBI DDRC,PINC0 // To show the status through LED

// Make PORTB0 to work as input pin
CBI DDRB,PINB0

// To enable Internal Pull-up network
SBI PORTB,PINB0
```

```
LDI R17,0x01
MAIN: CBI PORTC,PINC0
Loop: IN R16,PINB
      AND R16,R17
      BRNE MAIN
      SBI PORTC,PINC0
      JMP Loop
```

- **Make the below Circuit on SimulIDE to simulate**

Components Required:

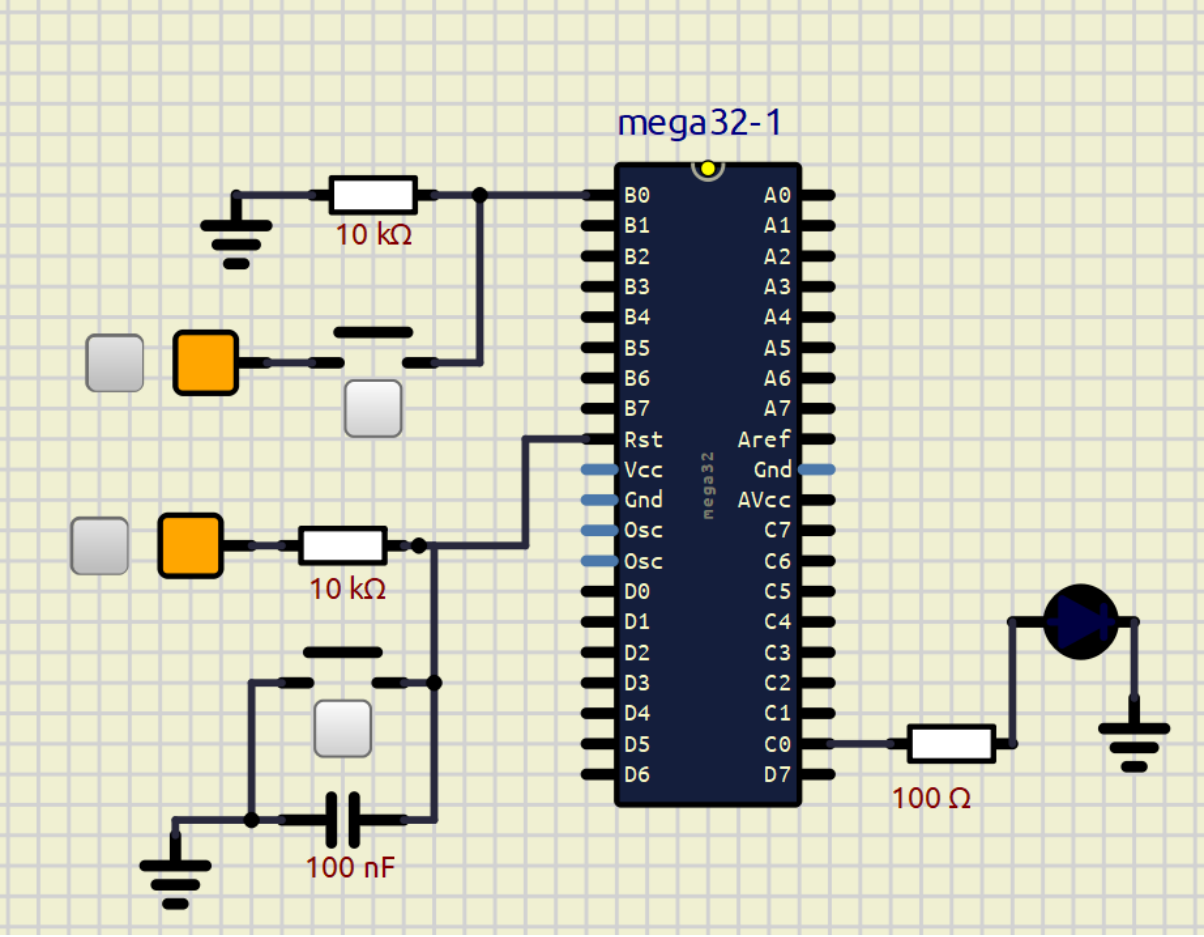
1. ATmega32
2. Push Switches
3. 10K Ohm Resistors
4. 100 Ohm Resistor
5. Fixed Voltage
6. 100nF Capacitor



After uploading the HEX file, verify the simulation by TA and Show the MCU Monitor Status.

Assignment2: Write the C code for the above Experiment

- **Class Assignment 1: Write an assembly code for ATMEGA32 as shown in below circuit to accept digital input through PULL-DOWN network. Simulate the code on SimulIDE and make the hardware.**



- **Class Assignment 2: Write an assembly code for ATMEGA32 as shown in below circuit to accept digital input through internal PULL_UP networks to increment and decrement the digit on seven segment display.**

  PUSH Button at PORTB0: For Increment the digit
  PUSH Button at PORTD2: For Decrement the digit
  After 9 the digit should be 0 when incrementing.
  After 0 the digit should be 9 when decrementing.

Use PORTA for Seven Segment Display with a single resistor at Common Cathode.

Simulate the code on SimulIDE and make the hardware.