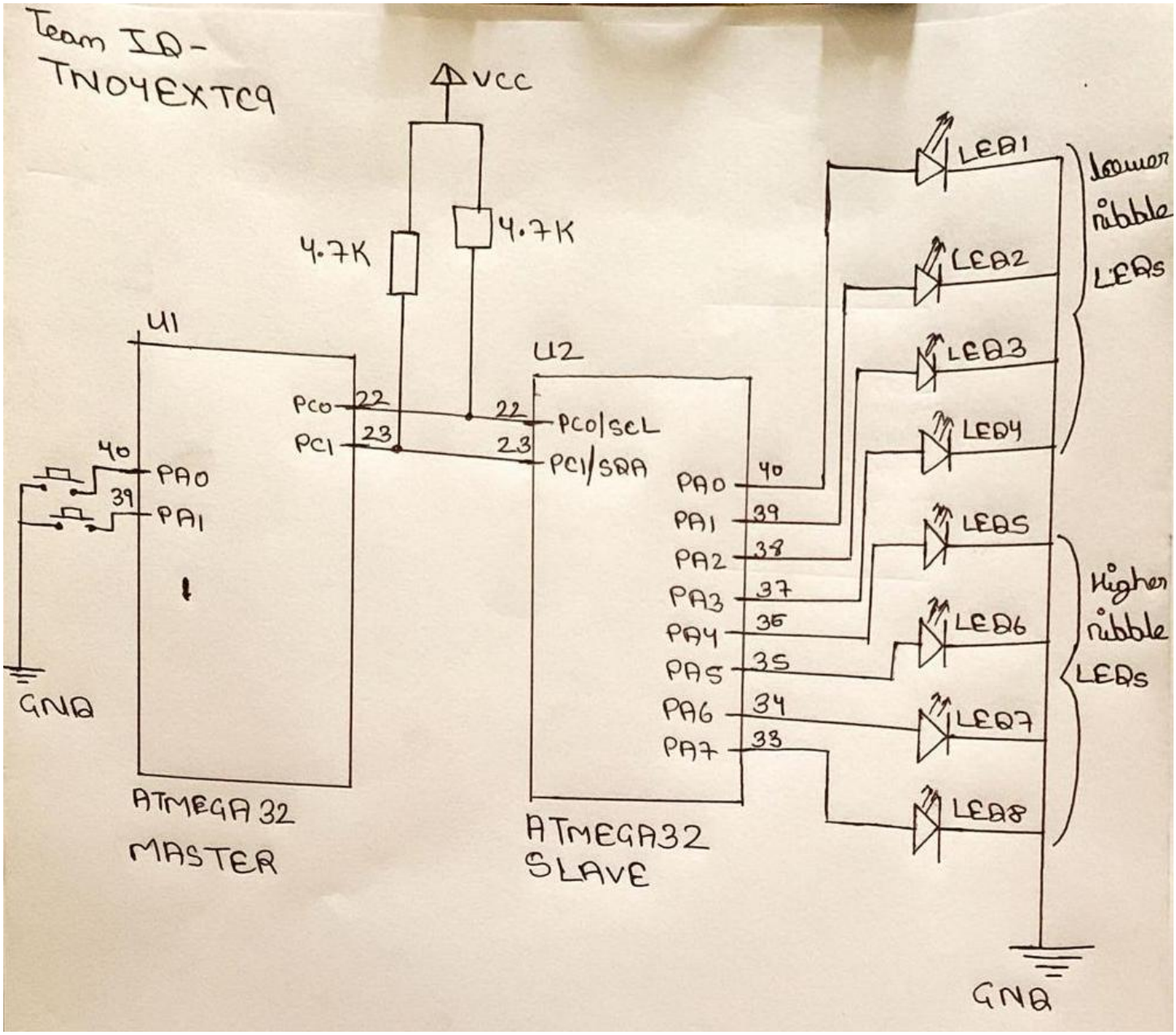
AVR I2C

Team Id: TN04ExTC9

Circuit Digarm for for Program:



- In Master Atmega32, there are Two Switchs are connected at PA0 and PA1. And In Slave Atmega32, there are 8 LED’s are connected at PA0 to PA7.
- The lower nibble LEDs are connected from PA0 to PA3, and The Higher nibble LEDs are connected from PA4 to PA7.
- When switch PA0 is pressed then Data (0x0F) willl be written in TWDR acoording to The Master Atmega32 program given below. The Lower nibble LEDs ON for 5 second at recevier side or Slvae Atmega32.
- When switch PA0 is pressed then Data (0xF0) willl be written in TWDR acoording to The Slave Atmega32 program given below. The Higher nibble LEDs ON for 5 second at recevier side or Slvae Atmega32.
- TWI only uses 2 bidirectional wires to establish communication among multiple devices. It can also adapt to the needs of various slave devices.

Program for Master Atmega32:

AVR I2C BUS TX - Microchip Studio

C:\Users\heman\Documents\Atmel Studio\7.0\AVR I2C BUS TX\AVR I2C BUS TX\main.c

```c
// Program for Master ATmega32

#define F_CPU 20000000                          // 20 MHz Clock Speed
#include <avr/io.h>
#include <util/delay.h>

#define Input_Switch DDRA
#define hight_Input PORTA

void TWI_Init()
{
    TWBR=0x01;                                  // Bit rate
    TWSR=(0<<TWPS1)|(0<<TWPS0);                 // Setting prescalar bits
                                                // SCL freq= F_CPU/(16+2(TWBR)4^TWPS)
}

void TWI_Start()
{
    TWCR = (1<<TWINT)|(1<<TWEN)|(1<<TWSTA);     // Clear TWI interrupt flag, Put start condition on SDA, Enable TWI
    while((TWCR & (1<<TWINT)) == 0);            // Wait till start condition is transmitted
    while((TWSR & (0xF8)) != 0x08);             // Check for the acknowledgement
}

void TWI_Write_Addr(unsigned char Addr)
{
    TWDR = Addr;                                // Address and write instruction
    TWCR = (1<<TWINT)|(1<<TWEN);                // Clear TWI interrupt flag,Enable TWI
    while((TWCR & (1<<TWINT)) == 0);            // Wait till complete TWDR byte transmitted
    while((TWSR & (0xF8)) != 0x18);             // Check for the acknoledgement
}

void TWI_Write_Data(unsigned char Data)
{
    TWDR = Data;                                // put data in TWDR
    TWCR = (1<<TWINT)|(1<<TWEN);                // Clear TWI interrupt flag,Enable TWI
    while((TWCR & (1<<TWINT)) == 0);            // Wait till complete TWDR byte transmitted
    while((TWSR & (0xF8)) != 0x28);             // Check for the acknoledgement
}

void TWI_Stop()
{
    TWCR = (1<<TWINT)|(1<<TWEN)|(1<<TWSTO);     // Clear TWI interrupt flag, Put stop condition on SDA, Enable TWI
    while((TWCR & (1<<TWSTO)) == 0);            // Wait till stop condition is transmitted
}

int main(void)
{
    Input_Switch &= ~((1<<PINA0)|(1<<PINA1));   // Making PA0 and PA1 as input pins for 2 Switchs
    hight_Input |= (1<<PINA0)|(1<<PINA1);       // Apply high input At PA0 and PA1
    TWI_Init();
    while(1)
    {
        TWI_Start();
        TWI_Write_Addr(0x20);
        if((PINA & (1<<PA0)) == 0)              // for a 1st switch pressed
        {
            TWI_Write_Data(0x0F);               // Lower nibble LED's on at receiver side for 5 Second
            _delay_ms(5000);
            TWI_Write_Data(0x00);               // Lower nibble LED's off at receiver side after 5 Second
        }
        if((PINA & (1<<PA1)) == 0)              // for a 2nd switch pressed
        {
            TWI_Write_Data(0xF0);               // Higher nibble LED's on at receiver side for 5 Second
            _delay_ms(5000);
            TWI_Write_Data(0x00);               // Higher nibble LED's off at receiver side after 5 Second
        }
        TWI_Stop();
    }
}
```

Output

Show output from: Build

```
        "C:\Program Files\Microchip\xc8\v2.31\bin\avr-objcopy.exe" -j .eeprom --set-section-flags=.eeprom=alloc,load --change-section-lma .eeprom
        "C:\Program Files\Microchip\xc8\v2.31\bin\avr-objdump.exe" -h -S "AVR I2C BUS TX.elf" > "AVR I2C BUS TX.lss"
    Done executing task "RunCompilerTask".
    Task "RunOutputFileVerifyTask"
            Program Memory Usage     :    300 bytes   0.9 % Full
            Data Memory Usage        :    0 bytes   0.0 % Full
            Warning: Memory Usage estimation may not be accurate if there are sections other than .text sections in ELF file
    Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "AVR I2C BUS TX.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Users\heman\Documents\Atmel Studio\7.0\A\
Done building target "Build" in project "AVR I2C BUS TX.cproj".
Done building project "AVR I2C BUS TX.cproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

Build succeeded

Program for Slave Atmega32:

AVR I2C BUS RX - Microchip Studio

```
// Program for Slave ATmega32

#define F_CPU 20000000                          // 20 MHz Clock Speed
#include <avr/io.h>
#include <util/delay.h>

#define LED_DDR DDRA
#define LED_PORT PORTA

void TWI_Init(unsigned char Slave_Addr)
{
    TWAR = Slave_Addr;                          // Fill slave address to TWAR
}
unsigned char TWI_Read_Data()
{
    unsigned char x;
    TWCR = (1<<TWINT)|(1<<TWEA)|(1<<TWEN);      // Clear TWI interrupt flag,Get acknowlegement, Enable TWI
    while((TWCR & (1<<TWINT)) == 0);            // Wait for TWINT flag
    while((TWSR & (0xF8)) != 0x80);             // Wait for acknowledgement
    x = TWDR;                                   // Get value from TWDR
    return x;
}
void TWI_Match_ACK()
{
    while((TWSR & (0xF8)) != 0x60)
    {
        TWCR = (1<<TWINT)|(1<<TWEA)|(1<<TWEN);  // Clear TWI interrupt flag, Get acknowlegement, Enable TWI,
        while((TWCR & (1<<TWINT)) == 0);        // Wait for TWINT flag
    }
}
int main(void)
{
    LED_DDR = 0xFF;                             // Making PORTA As Output port
    TWI_Init(0x20);
    while(1)
    {
        TWI_Match_ACK();
        LED_PORT = TWI_Read_Data();             // Send the receive value on PORTB
        _delay_ms(5000);                        // Lower or Higher nibble LED's on/off for 5 second
    }
}
```

```
Output
Show output from: Build
        C:\Program Files (x86)\Atmel\Studio\7.0\shellUtils\make.exe all --jobs 8 --output-sync
        make: Nothing to be done for 'all'.
    Done executing task "RunCompilerTask".
    Task "RunOutputFileVerifyTask"
                Program Memory Usage     :   202 bytes   0.6 % Full
                Data Memory Usage        :   0 bytes   0.0 % Full
                Warning: Memory Usage estimation may not be accurate if there are sections other than .text sections in ELF file
    Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "AVR I2C BUS RX.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Users\heman\Documents\Atmel Studio\7.0\A\
Done building target "Build" in project "AVR I2C BUS RX.cproj".
Done building project "AVR I2C BUS RX.cproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

Build succeeded