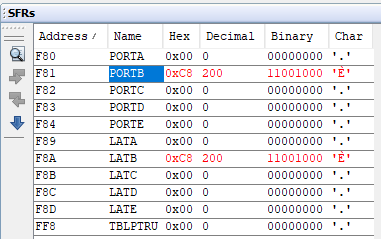
**Assignment 1 : simple addition program** #include<PIC18F458.h> #include<stdio.h>

void main(){ int a,b,c; a=100; b=100;

c=a+b; TRISB=0; PORTB=c;

}

## Output:

****

**Assignment 2 Addition of Array**

#include<PIC18F458.h> #include<stdio.h>

void main(){ int i,sum;

int n[]={11,12,13,14,15};

sum=0; for(i=0;i<=4;i++){

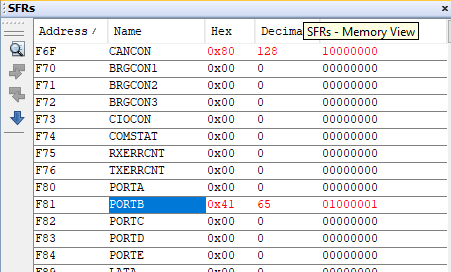
sum=sum+n[i];

} TRISB=0;

PORTB=sum;

}

## Output:

****

**Assignment 3 : Internal to internal memory transfer**

#include <stdio.h> #include <stdlib.h> #include <string.h> #include <PIC18F458.h> #pragma config OSC=HS #define SIZE 10

unsigned char data[SIZE];

unsigned char external\_mem[SIZE];

void transfer\_internal\_to\_internal(unsigned char \*src,unsigned char \*dest,int size)

{

memcpy(dest, src, size);

}

void transfer\_internal\_to\_external(unsigned char \*src,unsigned char \*dest,int size)

{

memcpy(dest, src, size);

}

void main(void)

{

for( int i = 0; i<SIZE; i++)

{

data[i]=i;

}

unsigned char internal\_data[SIZE]; transfer\_internal\_to\_internal(data,internal\_data,SIZE ); transfer\_internal\_to\_external(data,external\_mem,SIZE ); printf("Data in internal Memory : \n");

for (int i =0; i<SIZE; i++ ){ printf("%d",internal\_data[i]);

}

printf("\n");

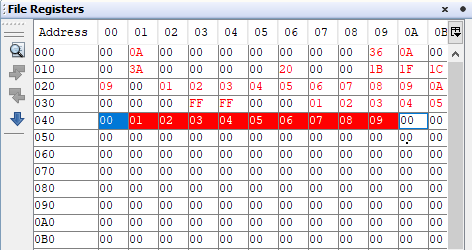
printf("Data in external Memory : \n"); for (int i =0; i<SIZE; i++ ){ printf("%d",external\_mem[i]);

}

printf("\n"); while(1);

}

## Output:

****

**Assignment 4: LED Interfacing**

#include <PIC18F458.h>

// Define the delay in milliseconds void delay();

void main(void) {

// Set the TRISB register to output mode for the LED TRISB = 0x00;

// Loop forever

// Turn the LED on while(1)

{

PORTB = 0xFF;

delay(); PORTB = 0x00;

delay();

}

}

void delay()

{

for(int i=0;i<=10000;i++)

{

}

}

## Output:

**Assignment 5: LCD Interfacing**

#include <PIC18F458.h>

#define ldata PORTB //define name ldata to PORTB

#define rs PORTCbits.RC1 //definne name rs(egister select) for RC0 pin of PORTC #define en PORTCbits.RC0 //definne name en(enable) for RC1 pin of PORTC

void delay();

void lcdcmd(unsigned char); void lcdval(unsigned char); void main()

{

unsigned char xc;

TRISB=0; //make PORTB as output TRISC=0; //make PORTC as output lcdcmd(0x38); //command to initialise LCD.

lcdcmd(0x0E); //command to make Display on cursor blinking lcdcmd(0x01); //command to clear display screen lcdcmd(0x06); //command to increment cursor

lcdcmd(0xC0); //command to set cursor at 1st line 6th position lcdval('H');

lcdval('E');

lcdval('L');

lcdval('L');

lcdval('O'); while(1);

}

void lcdcmd(unsigned char y)

{

rs=0; ldata=y; en=1; delay(); en=0; delay();

}

// lcdval fuction used to select DATA REGISTER of LCD by making,

// rs=1 and en=1 to 0 edge AND sends 8 bit command to PORTB void lcdval(unsigned char y)

{

rs=1; ldata=y; en=1; delay(); en=0; delay();

}void delay(){

for(int i=0;i<=2000;i++)

{}

}

## Output:

**Assignment 6 :DC Motor Interfacing**

#include<pic18f458.h> void main(void)

{

TRISCbits.RC2 = 0; // 00101100 to Select PWM mode; Duty cycle LSB CCP1CONbits.CCP1M = 0b1100; //CCP1CON<4:5> = <1:1>

TMR2ON = 0;

while(1)

{

PR2 = 74;

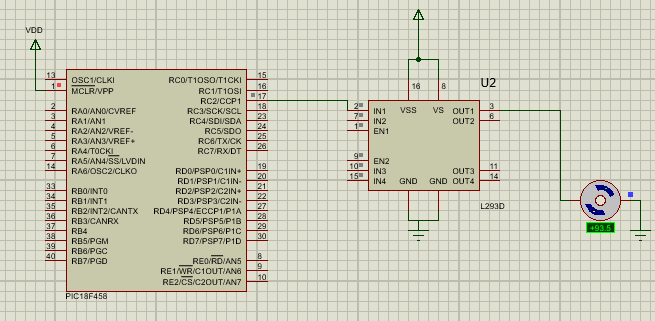
CCPR1L = 18;

TMR2ON = 1;

}

}

## Output:

****

**Assignment – 07 PC TO PC Serial Communication using UART**

#include <PIC18F458.h> void main()

{

unsigned char a[]="WELCOME"; ///character array to transmit int i;

TRISB=0; TRISCbits.RC6=0; TRISCbits.RC7=1;

SPBRG=0X4D;//to set baud rate RCSTAbits.SPEN = 1; // Enable USART module

TXSTAbits.TXEN = 1;//receive status and control register for(i=0;i<=6;i++)

{

while(PIR1bits.TXIF==0); //check whether data is transmitted or not TXREG=a[i];

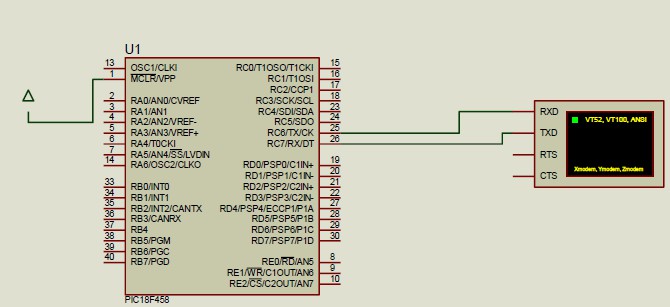
PIR1bits.TXIF=0;

}

while(1);

}

## Output:

****

**Assignment No. 8: Study of Arduino, Raspberry-Pi**

**Aim**: Study of Arduino, Raspberry-Pi, Beagle board.

**Theory**:

# A) Arduino Uno Board

**Arduino Uno** is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328.

First Arduino project was started in Interaction Design Institute Ivrea in 2003 by David Cuartielles and Massimo Banzi with the intention of providing a cheap and flexible way to students and professional for controlling a number of devices in the real world.

It allows the designers to control and sense the external electronic devices in the real world.

There are many versions of Uno boards available, however, Arduino Nano V3 and Arduino Uno arethe most official versions.

When nature and functionality of the task go complex, Mirco SD card can be added in the boards to make them store more information.

# Processor:

1. 8-bit AVR microcontroller Atmega328
2. 2KB SRAM and 1KB of EEPROM.
3. 13KB of flash memory.

# Features of Arduino Uno Board

1. Arduino Uno comes with USB interface i.e. USB port is added on the board to develop serial communication with the computer. Apart from USB, battery or AC to DC adopter can also be used to power the board. It is an open source platform where anyone can modify and optimize the board based on the number of instructions and task they want to achieve.
2. This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.
3. There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins.
4. The 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to the high range using analog Reference() function and AREF pin.
5. 13KB of flash memory is used to store the number of instructions in the form of code.
6. Only 5 V is required to turn the board on, which can be achieved directly using USB port or external adopter, however, it can support external power source up to 12 V which can be regulated and limit to 5 V or 3.3 V based on the requirement of the project.

# Pin Description

I/O digital and analog pins placed on the board which operates at 5V. These pins come with standard operating ratings ranging between 20mA to 40mA. Internal pull-up resistors are used in theboard that limits the current exceeding from the given operating conditions. However, too much increase in current makes these resisters useless and damages the device.

**LED:** Arduino Uno comes with built-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON and LOW will turn it OFF.

**Vin:** It is the input voltage provided to the Arduino Board. Vin and Power Jack support a voltage ranges between 7V to 20V. This pin is used to supply voltage. If a voltage is provided through power jack, it can be accessed through this pin.

**5V:** This board comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using three ways i.e. USB, Vin pin of the board or DC power jack.

**GND:** These are ground pins. More than one ground pins are provided on the board which can be used as per requirement.

**RESET:** This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.

**IOREF:** This pin is very useful for providing voltage reference to the board. A shield is used toread the voltage across this pin which then selects the proper power source.

**PWM:** PWM is provided by 3, 5, 6, 9, 10, 11 pins. These pins are configured to provide 8- bit output PWM.

**SPI:** It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of SPI library.

**AREF:** It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.

**TWI:** It is called Two-wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose.

**Serial Communication:** The Atmega328 placed on the board provides serial communication using pins like Rx and Tx. It is carried out through these pins called Pin 0 (Rx) and Pin 1 (Tx).

Rx pin is used to receive data while Tx pin is used to transmit data.

**External Interrupts:** Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

* 1. **Raspberry Pi Board**

**Raspberry Pi** is a credit card sized bargain micro-Linux machine. The goal behind creating Raspberry Pi was to create a low cost device that would improve programming skills

and hardware understanding for students.Raspberry Pi is open hardware with the exception of its primary chip, theBroadcomm SoC which runs the main components of the board – CPU, graphics, memory, USB controller etc.

## Processor:

1. The Model A+ is the low-cost variant of the Raspberry Pi. It has 256MB RAM, one USB port,40 GPIO pins and no Ethernet port.
2. The Model B+ is the final revision of the original Raspberry Pi. It has 512MB RAM, four USB ports, 40 GPIO pins, and an Ethernet port.
3. The Pi 2 uses a 900MHz quad-core ARM Cortex-A7 CPU and has 1GB RAM.
4. The Pi 3 Model B was launched in February 2016; it uses a 1.2GHz 64-bit quad- core ARMCortex-A53 CPU, has 1GB RAM, integrated 802.11n wireless LAN, and Bluetooth 4.1.
5. Pi Zero is half the size of a Model A+, with a 1Ghz single-core CPU and 512MB

RAM, andmini-HDMI and USB On-The-Go ports.

1. Operating Systems: Raspbian RaspBMC, Arch Linux, Rise OS, OpenELEC Pidora
2. Video Output: HDMI Composite RCA
3. Supported Resolutions: 640x350 to 1920x1200, including 1080p, PAL & NTSC standards
4. Power Source: Micro USB

## Hardware required to setup your Raspberry Pi

1. A Raspberry Pi
2. An HDMI or composite video capable television or monitor
3. An HDMI or composite video cable
4. An SD card that is compatible with your Raspberry Pi - <http://elinux.org/Raspberry> Pi\_SD\_cards has

a list of SD cards you should use

1. A USB keyboard and mouse (Bluetooth keyboard/mouse work for latest model but with minor connectivity issues)
2. Standard Ethernet cable
3. Micro USB power supply (that can provide at least 700mA at 5V)
4. A 3.5 mm stereo audio cable if your project requires Raspberry Pi to be connected to external speakers.

## Steps to setup Raspberry Pi

Each of the following steps is detailed in the subsequent slides. Step 1 : SD Card Setup

Step 2 : Raspberry Pi cabling

Step 3 : Booting your Raspberry Pi for the first time Step 4 : Load GUI environment to your Raspberry Pi Sep 5 : Setup a network connection

Step 6: After completing this setup, you will have your device powered up and working as full-fledged

Linux box running Debian.

## Raspberry Pi cabling

1. Push SD card into the SD card slot.
2. Plug the HDMI cable into the HDMI output of the Raspberry Pi and connect to the

TV/monitor.

1. Turn on monitor and switch to the HDMI port.
2. Insert the network cable and connect to the router.
3. Connect the keyboard and mouse via USB ports.
4. Plug the power supply into the micro USB.
5. The device is now ready for the next steps.

## Voltages

Two 5V pins and two 3V3 pins are present on the board, as well as a number of ground pins (0V),which are not configurable.

## Outputs

A GPIO pin designated as an output pin can be set to high (3V3) or low (0V).

## Inputs

A GPIO pin designated as an input pin can be read as high (3V3) or low (0V). This is made easier with the use of internal pull-up or pull-down resistors. Pins GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins this can be configured in software. The GPIO pins can be used with a variety of alternative functions; some are available on all pins, others on specific pins such as

1. PWM (pulse-width modulation)
2. Software PWM available on all pins
3. Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19
4. SPI
5. SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
6. SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17);CE2

(GPIO16)

1. I2C
2. Data: (GPIO2); Clock (GPIO3)
3. EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
4. Serial
5. TX (GPIO14); RX (GPIO15)

## Difference between Arduino and Raspberry Pi

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Raspberry Pi** | **Arduino** |
| 1 | It is a mini computer with Raspbian OS.It can run multiple programs at a time. | Arduino is a microcontroller, which is a part of the computer. It runs only one program again and again. |
| 2 | It is difficult to power using a batterypack. | Arduino can be powered using a battery pack. |
| 3 | It requires complex tasks like installing libraries and software for interfacing sensors and other components | It is very simple to interface sensors and other electronic components to Arduino. |

|  |  |  |
| --- | --- | --- |
| 4 | It is expensive | It is available for low cost. |
| 5 | Raspberry Pi can be easily connected to the internet using Ethernet port and USB Wi-Fi dongles. | Arduino requires external hardware to connect to the internet and this hardware is addressed properly using code. |
| 6 | Raspberry Pi did not have storage on board. It provides an SD card port. | Arduino can provide onboard storage. |
| 7 | Raspberry Pi has 4 USB ports to connect different devices. | Arduino has only one USB port to connectto the computer. |
| 8 | The processor used is from ARM family. | Processor used in Arduino is from AVR family Atmega328P |
| 9 | This should be properly shutdown otherwise  there is a risk of files corruption and software problems. | This is a just plug and play device. If power is connected it starts running the program and if disconnected it simply stops. |
| 10 | The Recommended programming language is  python but C, C++, Python,ruby are pre- installed. | Arduino uses Arduino, C/C++ |

* 1. **Beagle Bone Board**

**The Beagle Board** is a low-power open-source single-board computer produced by Texas Instruments in association with Digi-Key and Newark element14. Beagle Bone Black is a low- cost, open source, community-supported development platform for ARM CortexTM-A8 processor developers and hobbyists.

## Processor:

1. AM335x 1GHz ARM® Cortex-A8
2. 512MB DDR3 RAM
3. 4GB 8-bit eMMC on-board flash storage
4. 3D graphics accelerator
5. NEON floating-point accelerator
6. 2x PRU (PRU-Programmable Real Time Unit) 32-bit microcontrollers

## Connectivity

* 1. USB client for power & communications
  2. USB host
  3. Ethernet
  4. HDMI
  5. 2x 46 pin headers

## Software Compatibility

1. Debian
2. Android
3. Ubuntu
4. Cloud9 IDE on Node.js w/ BoneScript library
5. plus, much more

## Difference between Raspberry Pi and BeagleBone Black

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **BeagleBone Black** | **Raspberry PI** |
| 1 | Better interfaces with external sensors | Better graphics capabilities |
| 2 | Faster, newer, better supported processor | Better audio capabilities |
| 3 | Includes an OS out of the box | Better community support |
| 4 | Runs Angstrom, Ubuntu and other OS’s | Mainly runs Raspbian OS |

**Applications**

The devices discussed above come in wide range of applications. A larger number of people are using these boards for developing sensors and instruments that are used in scientific research. Following are some main applications of the board.

1. Embedded System
2. Security and Defense System
3. Digital Electronics and Robotics
4. Parking Lot Counter
5. Weighing Machines
6. Traffic Light Count Down Timer
7. Medical Instrument
8. Emergency Light for Railways
9. Home Automation
10. Industrial Automatio