**Assignment 1 – ARM7**

Q.1.) How many registers does the ARM 7 support? What are they for? Three of the registers each serve special purpose, what are those purposes?

Ans.1)ARM processors provide general-purpose and special-purpose registers. Some additional registers are available in privileged execution modes.

In all ARM processors, the following registers are available and accessible in any processor mode:

13 general-purpose registers R0-R12.

▪ One Stack Pointer (SP).

▪ One Link Register (LR).

▪ One Program Counter (PC).

▪ One Application Program Status Register (APSR).

ARM processors, with the exception of ARMv6-M and ARMv7-M based processors, have a total of 37 registers, with 3 additional registers if the Security Extensions are implemented, and in ARMv7-A only, 3 more if the Virtualization Extensions are implemented. The registers are arranged in partially overlapping banks. There is a different register bank for each processor mode. The banked registers give rapid context switching for dealing with processor exceptions and privileged operations.

The additional registers that are available in privileged software execution, with the exception of ARMv6-M and ARMv7-M, are:

▪ Two Supervisor mode registers for banked SP and LR.

▪ Two Abort mode registers for banked SP and LR.

▪ Two Undefined mode registers for banked SP and LR.

▪ Two Interrupt mode registers for banked SP and LR.

▪ Seven FIQ mode registers for banked R8-R12, SP and LR.

▪ Two Monitor mode registers for banked SP and LR. These are only present if the Security Extensions are implemented.

▪ Two Hyp mode registers for banked SP, and to hold the return address from Hyp mode. These are only present if the Virtualization Extensions are implemented.

▪ One Saved Program Status Register (SPSR) for each exception mode.

Q.2.) What are the major categories into which the ARM 7 ARM instructions are organized?

Ans 2.)Type of operation:

1. Arithmetic

2. Branch

3. Load and Store

4. Logical

5. Move

6. Alphabetic lists of instructions and mnemonics

Other features:

1. Condition fields

2. Addressing modes

3. Second operand

4. Shifts

Q.3.) Explain ARM programmer’s model in detail.

Ans 3.)

**Register File.**In the ARM processor, 16 general purpose registers are available at any time. Each register is 32-bit in size. The registers are referred to as rn, where n represents the register index. All instructions treat registers r0 to r13 equally. Any operation that can be performed on r0 can be performed equally well on registers r1 to r13. But r14 and r15 are assigned special functions by the processor. r15 is the program counter, and contains the address of the next instruction to be fetched. r14 is the link register, and used to store the return address, when a subroutine is invoked.

|  |
| --- |
|  |
|

**Current Program Status Register.**The Current Program Status Register (cpsr) is a dedicated 32-bit register, that contains the following fields.

1. Condition Flags
2. Interrupt Masks
3. Processor Mode
4. Processor State

Only the condition flags field will be used in the examples provided in this tutorial. And hence only the condition flags will be elaborated here.

The condition flags indicates the various conditions that occur while performing arithmetic and logical operations. The various condition flags and their meaning are given in the following table.

**Condition Flags**

| **Flag** | **Meaning** |
| --- | --- |
| Carry C | Operation caused a carry. |
| Overflow O | Operation caused an overflow. |
| Zero Z | Operation resulted in 0. |
| Negative N | Operation resulted in a negative value. |

Q.4.)Draw and explain ARM Core Data Flow Model.

Ans 4.)

• The Sign extend hardware converts signed 8-bit & 16-bit numbers to 32-bit values as they are read from memory & placed in a register (for signed values), fill zeros if unsigned.

• Source operands (Rn & Rm) are read from the register file using the internal buses A & B respectively & result Rd is written back.

• The PC value is in the address register which is fed in to the incrementer, then the incremented value is copied back in to r15. It is also written in to address register to be used as the address for the next instruction fetch.

• ALU: (The Arithmetic & logic Unit) or MAC (multiply & accumulate Unit) takes the register values Rn & Rm from A & B buses & computes the result.

• Barrel shifter:

• One important feature of the is that register Rm alternatively can be pre processed in barrel in barrel shifter before it enters the ALU [left shift, right shift, rotate etc.]

• Depending on the instruction Barrel Shifter may be used or it could be short circuit.

**Assignment 2 – Arduino and Raspberry Pi**

Q.1.)List the family of Arduino and explain any two in detail.

Ans1)The List Of Arduino are such as :-

* 1. Arduino Uno (R3)
  2. LilyPad Arduino
  3. Red Board
  4. Arduino Mega (R3)
  5. Arduino Leonardo

**Arduino UNO (R3) :-**

The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a [AC-to-DC adapter](https://www.pololu.com/product/1463) or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features an ATmega16U2 programmed as a USB-to-serial converter. This auxiliary microcontroller has its own USB bootloader, which allows advanced users to reprogram it.

The Arduino has a large [support community](http://arduino.cc/forum/) and an extensive set of support libraries and hardware add-on “[shields](https://www.pololu.com/category/113/arduino-shields)” (e.g. you can easily make your Arduino wireless with our [Wixel shield](https://www.pololu.com/product/2513)), making it a great introductory platform for embedded electronics. Note that we also offer a [SparkFun Inventor’s Kit](https://www.pololu.com/product/2776), which includes an Arduino Uno along with an assortment of components (e.g. breadboard, sensors, jumper wires, and LEDs) that make it possible to create a number of fun introductory projects.

**Arduino Leonardo :-**

The Arduino Leonardo is a microcontroller board based on the ATmega32U4. It has 23 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started (USB cable and power supply are not included). This board ships with the female header pins soldered in.

The Leonardo differs from all preceding Arduino boards in that the user-programmable ATmega32U4 AVR microcontroller has built-in USB functionality, eliminating the need for a secondary processor. This makes the Leonardo more versatile: in addition to supporting a virtual (CDC) serial/COM port interface, it can appear to a connected computer as a mouse and keyboard. See Arduino’s [getting started page](http://arduino.cc/en/Guide/ArduinoLeonardo) for more implications of the Leonardo’s single-MCU design.

The Arduino has a large [support community](http://arduino.cc/forum/) and an extensive set of support libraries and hardware add-on “[shields](https://www.pololu.com/category/113/arduino-shields)” (e.g. you can easily make your Arduino wireless with our [Wixel shield](https://www.pololu.com/product/2500)), making it a great introductory platform for embedded electronics. Note that we also offer a [SparkFun Inventor’s Kit](https://www.pololu.com/product/2776), which includes an Arduino Uno along with an assortment of components (e.g. breadboard, sensors, jumper wires, and LEDs) that make it possible to create a number of fun introductory projects.

Q.2.)Explain ADC module in Arduino Uno board.

Ans 2)

**Introduction**

●        When we interface sensors to the microcontroller, the output of the sensor many of the times is analog in nature. But microcontroller processes digital signals.

●        Hence, we use ADC in between sensor and microcontroller. It converts an analog signal into digital and gives it to the microcontroller.

●        There are many applications of ADC like in a biometric application, Environment monitoring, Gas leakage detection etc.

Arduino Uno has 6 0n-board ADC channels which can be used to read analog signal in the range 0-5V.

It has 10-bit ADC means it will give digital value in the range of 0 – 1023 (2^10). This is called as resolution which indicates the number of discrete values it can produce over the range of analog values.

**Digital Output value Calculation**

**ADC Resolution = Vref / ((2^n) - 1)**

**Digital Output = Vin / Resolution**

**Where,**

**Vref -** The reference voltage is the maximum value that the ADC can convert.

To keep things simple, let us consider that Vref is 5V,

For 0 Vin, digital o/p value = 0

For 5 Vin, digital o/p value = 1023 (10-bit)

For 2.5 Vin, digital o/p value = 512 (10-bit)

Functions for Arduino ADC

●        **analogRead (pin)**

This function is used to read analog value from specified analog pin.

**pin -** number of analog pin which we want to read

**returns -** digital value 0 – 1023

**e.g.** analogRead(A0) //read analog value at A0 channel

●        **analogReference (type)**

This function is used for configuring the reference voltage used for analog input.

Read Analog value using Arduino

Let’s write a program to read varying analog value generated using potentiometer which is connected to A0 analog channel. Display the digital value on Serial monitor which we got from the Arduino ADC.

**Sketch for reading analog value**

int sensorPin = A0;  // input pin for the potentiometer

int digitalValue = 0;// variable to store the value coming from the sensor

void setup() {

Serial.begin(9600);

}

void loop() {

digitalValue = analogRead(sensorPin);// read the value from the analog channel

Serial.print("digital value = ");

Serial.println(digitalValue);        //print digital value on serial monitor

delay(1000);

}

**Read Analog Voltage using Arduino Uno**

As ADC provide digital output which is proportional to analog value. To know what is input analog value, we need to convert this digital value back to analog value through program. To convert this digital value to analog input voltage,

**Aout = digital value \* (Vref/2^n – 1)**

**e.g.** digital value = 512 and ADC is 10-bit with 5V Vref. But, we want to know that for what analog voltage it is giving respective digital value. Then,

Aout = 512 \* (5 V / 1023)

= 2.5 V

Q.3.)Write complete steps to connect Raspberry Pi to a laptop using Ethernet cable.

Ans 3)Step 1: Setting Up Your Raspberry Pi

To connect raspberry pi to laptop display, you can simply use an ethernet cable. The desktop GUI (Graphical User Interface) of the raspberry pi can be viewed through the laptop display using a 100Mbps ethernet connection between the two. There are many softwares available that could establish connection between the raspberry pi and your laptop. We would be using the VNC server software to connect the pi to your laptop. Installing the VNC server on your pi allows you to see the raspberry pi’s desktop remotely, using the mouse and keyboard as if you were sitting right in front of your pi. It also means that you can put your pi anywhere else in your home, but still control it. Also, internet can be shared from laptop’s WiFi over Ethernet. This also lets you access internet on the pi and connect raspberry pi to laptop display.

Before moving to connect raspberry pi to laptop display, you need an SD card having the OS preinstalled. You will find lots of blogs and tutorials about preparing an SD card for the Raspberry Pi. But if you are a beginner, you can simply download this free beginner’s guide eBook on pi: Raspberry Pi guide. This will show how to install the OS for the raspberry pi.

After setting up your SD Card, insert it into the raspberry pi. Next, for powering the pi connect your micro USB cable to it. Also connect your raspberry pi to the laptop via an ethernet cable. And connect the keyboard & mouse to it. Now, connect the HDMI display (the HDMI is only required for running the pi for the first time). Now power on your Pi. And follow the next steps to connect raspberry pi to laptop display.

Step 2: Sharing Internet Over Ethernet

This step explains how you can share your laptop internet with the raspberry pi via Ethernet cable.In Windows : For sharing internet to multiple users over Ethernet, go to Network and Sharing Center. Then click on the WiFi network:

Click on Properties (shown below), then go to Sharing and click on “Allow other network users to connect”. Make sure that networking connection is changed to “Local Area Connection”:

Note : Doing this will provide a dynamic IP to the Ethernet port of your laptop and other Devices connected to your laptop. Now, for checking the IP assigned to your laptop, Click on the new local area connection link created:

As shown above, the IP assigned to my laptop is 192.168.137.1. For checking the IP assigned to the connected ethernet device, do the following. Considering that IP assigned to your Laptop is 192.168.137.1 and subnet mask is 255.255.255.0 :

Open command prompt.

Ping on broadcast address of your IP. (Type) Eg: ping 192.168.137.255.

Stop the ping after 5 seconds. Check the reply from device: arp –a

Step 3: Setting Up the VNC Server to Connect Raspberry Pi to Laptop Display

If you have an HDMI display: Using the connected HDMI display on your pi, you should install VNC server in raspberry pi. Open the LX-Terminal and type the following commands to install VNC:

$ sudo apt-get update

$ sudo apt-get install tight vncserver

If you don’t have an HDMI display: If you do not have a display even for one time setup, then no need to worry. Install Putty as per your windows configuration and via SSH you can connect with your raspberry pi. And, as you get access of your pi terminal, run the same commands as above to install VNC. Starting VNC Server on Pi : For starting VNC, enter the following command in SSH terminal:

$ vncserver :1

You will be prompted to enter and confirm a password. This will be asked only once, during first time setup. Enter an 8 digit password. Note that this is the password you will need to use, to connect to the Raspberry Pi remotely. You will also be asked if you want to create a separate “read-only” password – say no (n). Yippeee :)….The VNC server is now running on your Pi and so we can attempt to connect to it. But first we must switch to the laptop, from which we want to control the pi. Then setup a VNC client to connect to the pi.

Step 4: Setting Up the Client Side (Laptop)

Download VNC client from [here](http://www.realvnc.com/download/vnc/) and install it. When you first run VNC viewer.

Enter IP address of your raspberry pi given dynamically by your laptop (you got the address from the earlier step). And append with :1 (denoting port number) and press connect. You will get a warning message, press ‘Continue’

Enter the 8 digit password which was entered in VNC server installation on rpi.

Finally, the raspberry pi desktop itself should appear as a VNC window. You will be able to access the GUI and do everything, as if you were using the pi’s keyboard, mouse and monitor directly. As with SSH, since this is working over your network, your pi could be situated anywhere as long as it is connected to your network.

So, whenever you want to do something with your pi, just connect it with ethernet cable to your laptop and power it. Then open VNCViewer, mention the IP address of your pi. And you can use the display of your laptop as the raspberry pi’s monitor.