**9 June 2020**

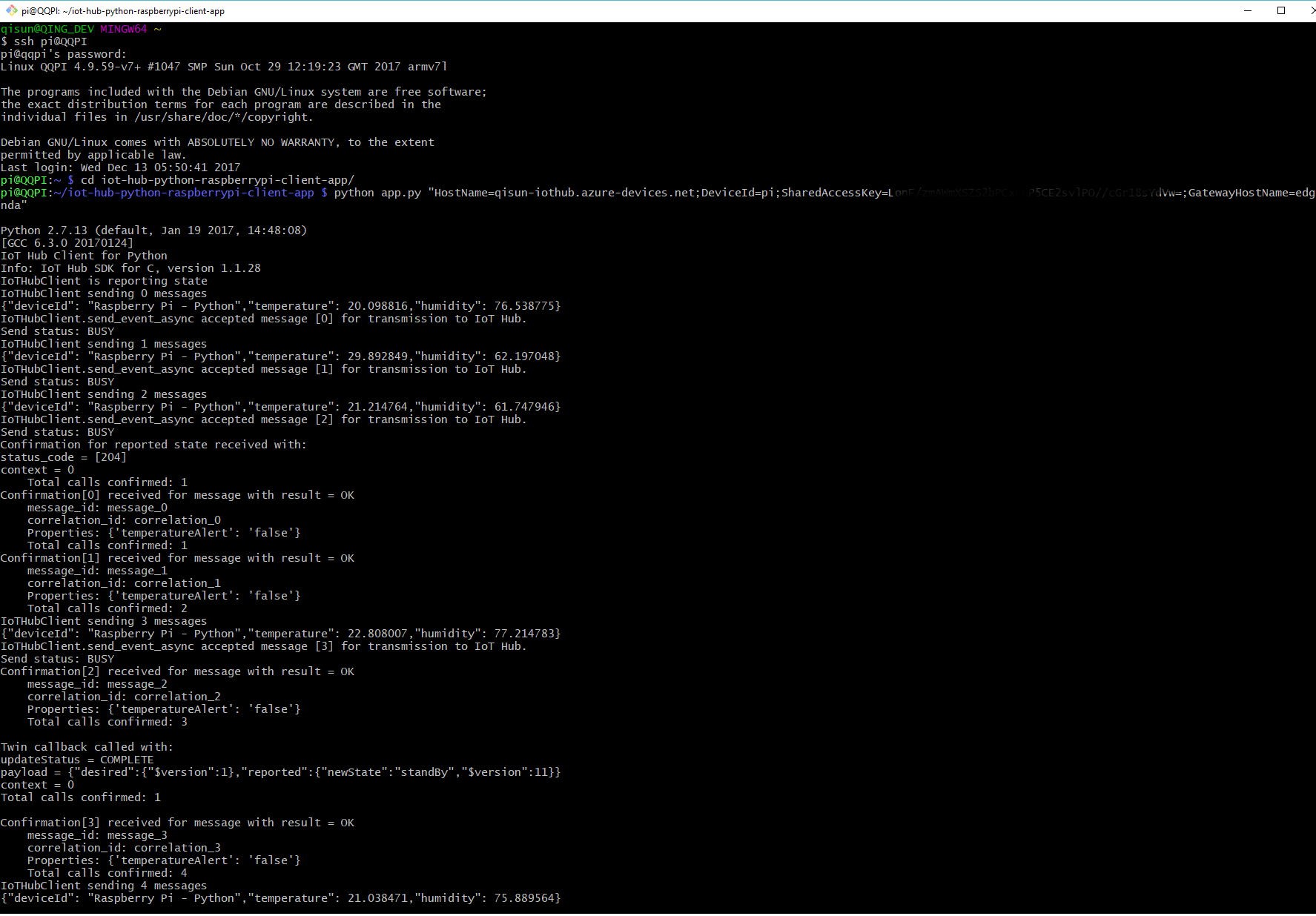
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date:** | | **9 June 2020** | | | **Name:** | **Srinidhi J C** |
| **Course:** | | **Logic Design** | | | **USN:** | **4al16ec078** |
| **Topic:** | | **MOS Transistor Basics - 1** | | | **Semester & Section:** | **8th -Sem, B-Sec** |
| **Github Repository:** | | **SrinidhiJC078** | | |  |  |
| **FORENOON SESSION DETAILS** | | | | | | |
| **Image of session**  **A screenshot of a cell phone  Description automatically generatedA close up of text on a white background  Description automatically generatedA screenshot of a cell phone  Description automatically generatedA screenshot of a cell phone  Description automatically generatedA screenshot of a cell phone  Description automatically generatedA screenshot of a cell phone  Description automatically generatedA screenshot of a cell phone  Description automatically generatedA screenshot of a cell phone  Description automatically generated** | | | | | | |
| **Report**  MOSFET (CMOS) technology is widely used today to form circuits in numerous and varied applications. Today’s computers, CPUs and cell phones make use of CMOS due to several key advantages. CMOS offers low power dissipation, relatively high speed, high noise margins in both states, and will operate over a wide range of source and input voltages (provided the source voltage is fixed)  For the processes we will discuss, the type of transistor available is the Metal-Oxide-Semiconductor Field Effect Transistor (MOSFET). These transistors are formed **as a ‘sandwich’** consisting of a semiconductor layer, usually a slice, or wafer, from a single crystal of silicon; a layer of silicon dioxide (the oxide) and a layer of metal.  **Structure of a MOSFET**  As shown in the figure, MOS structure contains three layers −   * **The Metal Gate Electrode** * **The Insulating Oxide Layer (SiO2)** * **P – type Semiconductor (Substrate)**   MOS structure forms a capacitor, with gate and substrate are as two plates and oxide layer as the dielectric material. The thickness of dielectric material (SiO2) is usually between 10 nm and 50 nm. Carrier concentration and distribution within the substrate can be manipulated by external voltage applied to gate and substrate terminal. Now, to understand the structure of MOS, first consider the basic electric properties of P – Type semiconductor substrate.  Concentration of carrier in semiconductor material is always following the **Mass Action Law**. Mass Action Law is given by −  $$n.p=n\_{i}^{2}$$  Where,   * **n** is carrier concentration of electrons * **p** is carrier concentration of holes * **ni** is intrinsic carrier concentration of Silicon   Now assume that substrate is equally doped with acceptor (Boron) concentration NA. So, electron and hole concentration in p–type substrate is  $$n\_{po}=\frac{n\_{i}^{2}}{N\_{A}}$$  $$p\_{po}=N\_{A}$$  Here, doping concentration **NA** is (1015 to 1016 cm−3) greater than intrinsic concentration ni. Now, to understand the MOS structure, consider the energy level diagram of p–type silicon substrate.  P-type Silicon Substrate  As shown in the figure, the band gap between conduction band and valance band is 1.1eV. Here, Fermi potential ΦF is the difference between intrinsic Fermi level (Ei) and Fermi level (EFP).  Where Fermi level EF depends on the doping concentration. Fermi potential ΦF is the difference between intrinsic Fermi level (Ei) and Fermi level (EFP).  Mathematically,  $$\Phi\_{Fp}=\frac{E\_{F}-E\_{i}}{q}$$  The potential difference between conduction band and free space is called electron affinity and is denoted by qx.  So, energy required for an electron to move from Fermi level to free space is called work function (qΦS) and it is given by  $$q\Phi \_{s}=(E\_{c}-E\_{F})+qx$$  The following figure shows the energy band diagram of components that make up the MOS.  Energy Level Diagram of Components  As shown in the above figure, insulating SiO2 layer has large energy band gap of 8eV and work function is 0.95 eV. Metal gate has work function of 4.1eV. Here, the work functions are different so it will create voltage drop across the MOS system. The figure given below shows the combined energy band diagram of MOS system.  Combined Energy Band Diagram  As shown in this figure, the fermi potential level of metal gate and semiconductor (Si) are at same potential. Fermi potential at surface is called surface potential ΦS and it is smaller than Fermi potential ΦF in magnitude.  **Working of a MOSFET**  MOSFET consists of a MOS capacitor with two p-n junctions placed closed to the channel region and this region is controlled by gate voltage. To make both the p-n junction reverse biased, substrate potential is kept lower than the other three terminals potential.  If the gate voltage will be increased beyond the threshold voltage (VGS>VTO), inversion layer will be established on the surface and n – type channel will be formed between the source and drain. This n – type channel will carry the drain current according to the VDS value.  For different value of VDS, MOSFET can be operated in different regions as explained below.  **Linear Region**  At VDS = 0, thermal equilibrium exists in the inverted channel region and drain current ID = 0. Now if small drain voltage, VDS > 0 is applied, a drain current proportional to the VDS will start to flow from source to drain through the channel.  The channel gives a continuous path for the flow of current from source to drain. This mode of operation is called **linear region**. The cross sectional view of an n-channel MOSFET, operating in linear region, is shown in the figure given below.  Linear Region  **At the Edge of Saturation Region**  Now if the VDS is increased, charges in the channel and channel depth decrease at the end of drain. For VDS = VDSAT, the charges in the channel is reduces to zero, which is called **pinch – off point**. The cross sectional view of n-channel MOSFET operating at the edge of saturation region is shown in the figure given below.  Edge of Saturation Region  **Saturation Region**  For VDS>VDSAT, a depleted surface forms near to drain, and by increasing the drain voltage this depleted region extends to source.  This mode of operation is called **Saturation region**. The electrons coming from the source to the channel end, enter in the drain – depletion region and are accelerated towards the drain in high electric field.  Saturation Region  **MOSFET Current – Voltage Characteristics**  To understand the current – voltage characteristic of MOSFET, approximation for the channel is done. Without this approximation, the three dimension analysis of MOS system becomes complex. The **Gradual Channel Approximation (GCA)** for current – voltage characteristic will reduce the analysis problem.  **Gradual Channel Approximation (GCA)**  Consider the cross sectional view of n channel MOSFET operating in the linear mode. Here, source and substrate are connected to the ground. VS = VB = 0. The gate – to – source (VGS) and drain – to – source voltage (VDS) voltage are the external parameters that control the drain current ID.  Gradual Channel Approximation  The voltage, VGS is set to a voltage greater than the threshold voltage VTO, to create a channel between the source and drain. As shown in the figure, x – direction is perpendicular to the surface and y – direction is parallel to the surface. | | | | | | |  |
| **Date:** | **9 June 2020** | | **Name:** | **Srinidhi J C** | | |
| **Course:** | **IOT in Python with Rosberry Pi** | | **USN:** | **4al16ec078** | | |
| **Topic:** |  | | **Semester & Section:** | **8th-Sem, B-Sec** | | |
| **AFTERNOON SESSION DETAILS** | | | | | | |

**Images os the session**

****

**A screenshot of a cell phone

Description automatically generatedA close up of a logo

Description automatically generated**

**Report:**

Create an edge device for gateway and another edge device for raspberry pi, here I name the gateway panda.

Generate all necessary certificates in gateway following this [guidance](https://docs.microsoft.com/en-us/azure/iot-edge/how-to-create-transparent-gateway).

git clone https://github.com/azure/azure-iot-sdk-c

cd azure-iot-sdk-c\tools\CACertificates

Set-ExecutionPolicy -ExecutionPolicy Unrestricted

. ./ca-certs.ps1

Test-CACertsPrerequisites

New-CACertsCertChain

New-CACertsEdgeDevice myGateway

Go to Azure portal Azure IoT Hub and navigate to Certificates, add a new certificate, providing the root CA file when prompted

Start the IoT Edge runtime (Windows):

iotedgectl setup --connection-string {device connection string}

--edge-hostname {gateway hostname}

--device-ca-cert-file {full path}/certs/myGateway-public.pem

--device-ca-chain-cert-file {full path}/certs/myGateway-all.pem

--device-ca-private-key-file {full path}/private/myGateway-private.pem

--owner-ca-cert-file {full path}/RootCA.pem

### Raspberry Pi

* Install [Raspbian] (<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-raspberry-pi-kit-python-get-started#set-up-raspberry-pi>)
* Optional: If you have the sensor in hand, wire them up following the link above, otherwise we could just simulate the data

Plugin a real screen to you raspberry pi or just ssh it.

Generate .crt from the PEM file you generated when step the gateway. Install it to Raspberry Pi, commands for reference:

openssl x509 -in RootCA.pem -inform PEM -out RootCA.crt

sudo cp RootCA.crt /use/local/share/ca-certificates/RootCA.crt

sudo update-ca-certificates

Run the app.py with device connectionstring appended with GatewayHostName, e.g.:

python app.py "HostName=qisun-iothub.azure devices.net; DeviceId=pi; SharedAccessKey=XXXXX; GatewayHostName=<gateway host name>"

**Drawing Picture with Turtles**

import turtle

#create window and turtle

window = turtle.Screen()

babbage = turtle.Turtle()

#draw stem and centre

babbage.left(90)

babbage.forward(100)

babbage.right(90)

babbage.circle(10)

#draw first petal

babbage.left(15)

babbage.forward(50)

babbage.left(157)

babbage.forward(50)

#tidy up window

window.exitonclick()

**Using Loops**

#draw second petal

babbage.left(15)

babbage.forward(50)

babbage.left(157)

babbage.forward(50)

#draw all petals

for i in range(1,24):

babbage.left(15)

babbage.forward(50)

babbage.left(157)

babbage.forward(50)

window.exitonclick()

A screenshot of a cell phone

Description automatically generated

import turtle

#create window and turtle

window = turtle.Screen()

babbage = turtle.Turtle()

#draw stem

babbage.color("green", "black")

babbage.left(90) babbage.forward(100)

babbage.right(90)

#draw centre babbage.color("black", "black")

babbage.begin\_fill()

babbage.circle(10)

babbage.end\_fill()