TEMPERATURE MONITORING SYSTEM

# Project Scenario

AAARGH! My BUTTERFLIES are getting COLD!

Let's imagine a scenario where you have a bunch of butterflies at your home as your baby pet! You are taking care of them every day, but they are just not feeling it and are not flying at all. Then you Google it up to find out what could be the reason behind this. After a worth of hustle and bustle you finally stumbled upon the fact that 'butterflies do not fly when they are cold!'

Now, to take better care of 'your precious', you set up a temperature monitoring system so that you get an alert whenever the temperature goes out of the ideal range to fly i.e. 22.78 degree Celsius to 37.78 degree Celsius.

Being a cold-blooded species, butterflies cannot regulate their body temperature and as a result, the surrounding air temperature has a major impact on their ability to function. So, they cannot fly if it's too cold or too hot!

**FACT:** If the temperature falls below 12.78 degree Celsius, butterflies are rendered immobile and unable to flee from their predators. But you will not let that happen because you love them! Right?

## Hardware Connections

* Bolt Wi-Fi Module
* LM35 Temperature Sensor
* Three Female-Male wires
* USB cable
* Power Adapter

## Software, Apps and Online Services

* Bolt Cloud
* Linux Operating System (Ubuntu)
* Telegram Messaging Application

## System Requirements

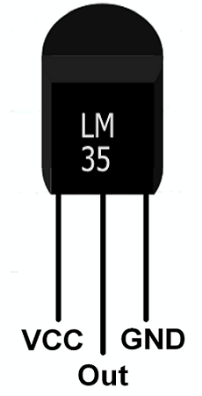
* A compatible 64-bit x86/AMD64 CPU launched in 2011 or later \*
* 1.3GHz or faster core speed
* 2GB RAM minimum/ 4GB RAM or more recommended

## Virtual OS: VMware

A **Virtual Machine** (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps. One or more virtual “guest” machines run on a physical “host” machine.  Each virtual machine runs its own operating system and functions separately from the other VMs, even when they are all running on the same host. This means that, for example, a virtual MacOS virtual machine can run on a physical PC. Virtual machine technology is used for many use cases across on-premises and cloud environments. More recently, public cloud services are using virtual machines to provide virtual application resources to multiple users at once, for even more cost efficient and flexible compute.

VMware Workstation Player is an ideal utility for running a single virtual machine on a Windows or Linux PC. Organizations use Workstation Player to deliver managed corporate desktops, while students and educators use it for learning and training.

# Sensor: LM35

LM35 is a temperature measuring device having an analog output voltage proportional to the temperature. It provides output voltage in Centigrade (Celsius). It does not require any external calibration circuitry. The sensitivity of LM35 is 10 mV/degree Celsius. As temperature increases, output voltage also increases. Example 250 mV means 25°C. It is a 3-terminal sensor used to measure surrounding temperature ranging from -55 °C to 150 °C. LM35 gives temperature output which is more precise than thermistor output. **VCC:**Supply Voltage (4V – 30V)

**FEATURES**

* Calibrated Directly in Celsius (Centigrade)
* Linear + 10-mV/°C Scale Factor
* 0.5°C Ensured Accuracy (at 25°C)
* Rated for Full −55°C to 150°C Range
* Suitable for Remote Applications
* Low-Cost Due to Wafer-Level Trimming
* Operates From 4 V to 30 V
* Less Than 60-μA Current Drain
* Low Self-Heating, 0.08°C in Still Air
* Non-Linearity Only ±¼°C Typical
* Low-Impedance Output, 0.1 Ω for 1-mA Load

# Telegram

Telegram is a popular cross-platform messaging app that is widely used because it offers some enhanced privacy and encryption features as well as support for large group chat features. It also has no ties to other social media platforms. The app is multiplatform, with versions available for iOS, Android, Windows, Mac, and Linux. It can also be accessed from a web browser. Telegram was founded by Russian social media entrepreneur Pavel Durov and the service is free to use.

**PROS**

* **End-to-end encryption**: When using the Secret Chat mode, communication is completely encrypted from end to end, rendering communication essentially bulletproof. You can also get end-to-end encryption in apps like WhatsApp and Signal.
* **Self-destructing messages**: Secret messages can be configured to self-destruct after a set period of time, making it even more secure. This is similar to what you can do in apps like Snapchat, Instagram, and Facebook Messenger.
* **Large file sizes**: Telegram supports file attachments up to 2 GB in size. This is one area where Telegram has virtually all other messaging apps beat. Only Skype comes even remotely close, limiting you to 300 MB files. Most apps are much more limiting - WhatsApp is just 16 MB, for example.

**CONS**

* **Limited user base**: While Telegram has several hundred million active monthly users, it's still significantly smaller than the 1 billion-plus active user bases on WhatsApp or Facebook Messenger. That means there's a good chance your friends and contacts are not using the app.
* **New user announcements may violate privacy**: One of the main reasons many people join Telegram is to be able to send secure and private messages. That's why it's frustrating that your contacts already on the app get notified when you join.
* **It's a gathering place for conspiracy theorists and hate groups**: As mainstream social media services increasingly take down QAnon groups, neo-Nazis, and other hate groups and conspiracy theorists, Telegram has become an attractive gathering place for many of these groups because of the enhanced security and permissive content policies, though Telegram has ramped up efforts to disable accounts run by hate groups and extremists.

# Setup

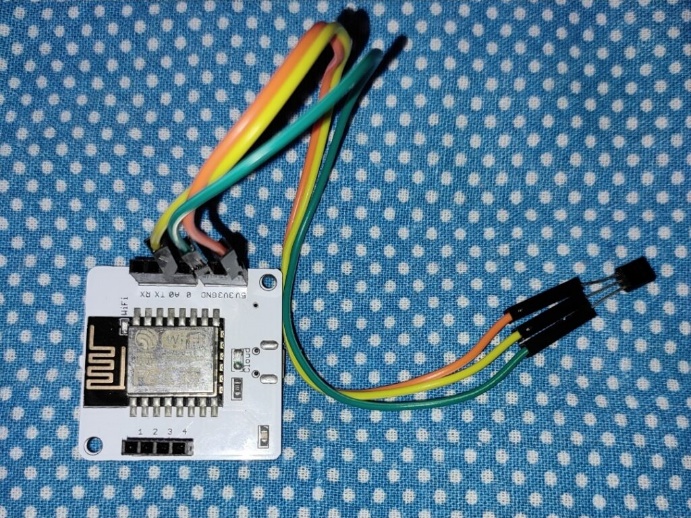
# LM35 Temperature SensorFemale-Male wiresBolt Wi-Fi Module Bolt Wi-Fi Module LM35 Temperature Sensor Female-Male Wire

**Step 1:** Setup your Bolt Wi-Fi module.

Register your Bolt Wi-Fi Module on Bolt Application.

1. It should show a stable blue and green light, indicating a proper connection.
2. Login to Bolt Cloud.
3. Here, note down the Device ID and API Key of your Bolt Wi-Fi Module that are present under the Devices and API tabs, respectively.

**NOTE:** Keep the Device ID and API Key somewhere safe, as they have to be used later on in the programming section.

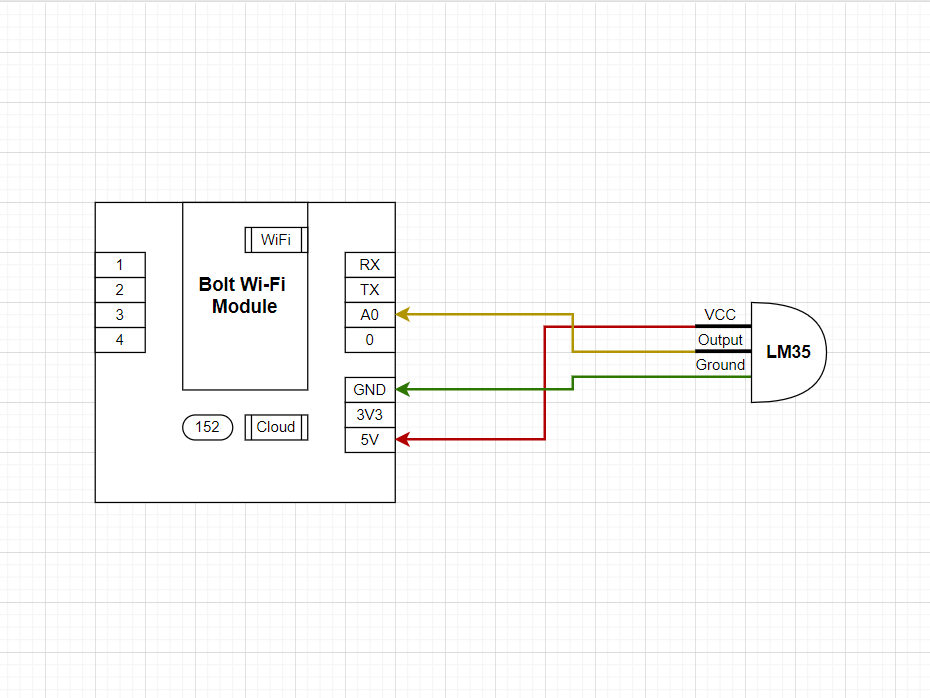
**Step 2:** Make the hardware connections.

**NOTE:** Make sure to do the connections before turning ON the power supply for Bolt Wi-Fi Module.

1. Take the LM35 temperature sensor. The written part should be facing towards you, to ensure proper connections.
2. Starting from Left to Right connect the pins of the sensor to your Bolt Module using Female-Male wires,
   * VCC to 5V
   * Output to A0
   * Ground to GND

**Safety Precaution:** Double check your connections so that there are no loose ends, as it can cause harm to you, the sensor and your module.

## Schematic Diagram



# Implementation

**Step 1:** Setup Telegram Channel

1. Open Telegram application and click on the "Pencil" icon in the bottom-right corner.
2. Click on "New Channel".
3. Enter a Name and Description for your channel.
4. In the next screen, set the channel as "Public" and enter a permanent link for your channel.
5. This link will be used later on to send alerts to this channel.

**Step 2:** Create a Bot

1. Click on the "Magnifying Glass" icon to search for the bot named "BotFather".
2. Click on the verified account of BotFather and type "**/start**" and send the message.
3. Then type "**/newbot**" to create a new bot.
4. Follow the instructions specified in the chat and enter the Bot name and Bot username.
5. After successfully creating a bot, it will provide you with A Token to access the HTTP API. Save it, as it will be used later on the the python code.

**Step 3:** Link the Telegram Channel and the Bot.

1. Open your new Telegram Channel. Click on the channel name and go to Administrators tab.
2. Here, search for your newly created bot and select it.

**NOTE:** You have successfully created a telegram channel and a bot for your temperature monitoring system.

**Step 4:** Setup the Bolt Python Library

1. Open your Ubuntu Server. It can be VMware, Digital Ocean or Virtual Machine Box. You can also open the terminal in Ubuntu Operating System.

**NOTE:** In case you do not have these software or Operating System then you can download them for free on the internet.

2. Type in the following commands.

sudo apt-get -y update #update packages on Ubuntu

sudo apt install python3-pip #install python3 pip3

sudo pip3 install boltiot #install boltiot library using pip

## Reading Sensor Values from Bolt Device

* 1. Create a new directory.

sudo mkdir temp\_alert #new directory

cd temp\_alert #move into directory

* 1. Create new file to hold the configurations of boltiot and Telegram.

sudo nano conf.py #new file

bolt\_api\_key = "XXXX"

# This is your Bolt Cloud API Key

device\_id = "XXXX"

# This is the device ID and will be similar to BOLTXXXX where XXXX is some numbers

telegram\_chat\_id = "@XXXX"

# This is the channel ID of the created Telegram channel. Paste after @ symbol.

telegram\_bot\_id = "botXXXX"

# This is the bot ID of the created Telegram Bot. Paste after bot text.

max\_threshold = 387

min\_threshold = 233

# Threshold beyond which the alert should be sent

**NOTE:** To save and exit from file, press "CTRL+X" and then press "Y". Hit "ENTER".

* 1. Create a new python file.

sudo nano telegram\_alert.py #new file

import requests # for making HTTP requests

import json # library for handling JSON data

import time # module for sleep operation

from boltiot import Bolt # importing Bolt from boltiot module

import conf # config file

mybolt = Bolt(conf.bolt\_api\_key, conf.device\_id)

def get\_sensor\_value\_from\_pin(pin):

"""Returns the sensor value. Returns -999 if request fails"""

try:

response = mybolt.analogRead(pin)

data = json.loads(response)

if data["success"] != 1:

print("Request not successfull")

print("This is the response->", data)

return -999

sensor\_value = int(data["value"])

return sensor\_value

except Exception as e:

print("Something went wrong when returning the sensor value")

print(e)

return -999

## Send Message via Telegram

Append this code to "telegram\_alert.py" to send message via Telegram.

def send\_telegram\_message(message):

"""Sends message via Telegram"""

url = "https://api.telegram.org/" + conf.telegram\_bot\_id + "/sendMessage"

data = {"chat\_id": conf.telegram\_chat\_id, "text": message}

try:

response = requests.request("POST", url, params=data)

print("This is the Telegram URL")

print(url)

print("This is the Telegram response")

print(response.text)

telegram\_data = json.loads(response.text)

return telegram\_data["ok"]

except Exception as e:

print("An error occurred in sending the alert message via Telegram")

print(e)

return False

## Compare Sensor Values with Threshold and Sending Alert Messages

Append this code to "telegram\_alert.py" to check if the temperature is within the range. If not, sends alert.

while True:

# Step 1

sensor\_value = get\_sensor\_value\_from\_pin("A0")

print("The current sensor value is: ", sensor\_value)

sensor\_value\_temp=(100\*sensor\_value)/1024

print("The current temperature is: ",sensor\_value\_temp)

# Step 2

if sensor\_value == -999:

print("Request was unsuccessfull. Skipping.")

time.sleep(10)

continue

# Step 3

if sensor\_value >= conf.max\_threshold:

print("Sensor value has exceeded the maximum threshold")

message = "Alert! Sensor value has exceeded the maximum threshold of " + str(conf.max\_threshold) + \

". The current value is " + str(sensor\_value)+\

". Your butterflies are at a temperature of

"+str(sensor\_value\_temp)+\". Save them!"

telegram\_status = send\_telegram\_message(message)

print("This is the Telegram status:", telegram\_status)

if sensor\_value >= conf.min\_threshold:

print("Sensor value has gone below the minimum threshold")

message = "Alert! Sensor value has gone below the minimum threshold of " + str(conf.min\_threshold) + \

". The current value is " + str(sensor\_value)+\

". Your butterflies are at a temperature of

"+str(sensor\_value\_temp)+\". Save them!"

telegram\_status = send\_telegram\_message(message)

print("This is the Telegram status:", telegram\_status)

# Step 4

time.sleep(10)

**NOTE:** Save and exit by pressing "CTRL+X" and then "Y". Hit "ENTER".

**Step 5:** Run the code.

sudo python3 telegram\_alert.py

## Output

