Simple Raspberry Pi Control With .NET IoT and Python

Follow the “Prerequisites” and “Prepare the hardware” instructions here. (Note: These instructions specify .NET SDK 5 or higher. We’ll actually be using .NET 6.)

Assumes that you’re using Linux on your development machine as well.

.NET

On the development machine, create a console application:

```
dotnet new console -o BlinkTutorial

cd BlinkTutorial
```

Add the Iot.Device.Bindings package to the project:

```
dotnet add package Iot.Device.Bindings --version 1.5.0-*
```

Replace the contents of Program.cs with the following code:

Program.cs

```
using System;
using System.Device.Gpio;
using System.Threading;

Console.WriteLine("Blinking LED. Press Ctrl+C to end.");
int pin = 18;
using var controller = new GpioController();
controller.OpenPin(pin, PinMode.Output);
bool ledOn = true;
while (true)
{
    controller.Write(pin, ((ledOn) ? PinValue.High : PinValue.Low));
    Thread.Sleep(1000);
    ledOn = !ledOn;
}
```

Make sure the application builds without errors:

```
dotnet build
```

Publish it:

```
dotnet publish -c Release -r linux-arm --self-contained true
/p:PublishSingleFile=true
```

Copy the application to the Raspberry Pi (adjust the remote machine name and path as needed):

```
scp -r bin/Release/net6.0/linux-arm/publish/* pi@raspi4-
main:/home/pi/projects/BlinkTutorial
```

Log in to the Raspberry Pi, go to the publish directory, and run the application:

```
ssh pi@raspi4-main

cd projects/BlinkTutorial

./BlinkTutorial
```

Enjoy the blinking light!

Makefile, to simplify the steps:

Makefile

```
REMOTE_USER_MACHINE = pi@raspi4-main

default:
    @echo 'Targets:'
    @echo '  build'
    @echo '  publish'
    @echo '  copy'
    @echo '  ssh'

build:
    dotnet build

publish:
    dotnet publish -c Release -r linux-arm --self-contained true
/p:PublishSingleFile=true

copy:
    scp -r bin/Release/net6.0/linux-arm/publish/*
$(REMOTE_USER_MACHINE):/home/pi/projects/BlinkTutorial

ssh:
    ssh $(REMOTE_USER_MACHINE)
```

Python

Log in to the Raspberry Pi:

```
ssh pi@raspi4-main
```

Create a directory for the Python script:

```
mkdir blink_tutorial

cd blink_tutorial
```

Install the gpio packages:

```
sudo apt-get install python-rpi.gpio python3-rpi.gpio
```

Create the script:

blinking_led.py

```
#!/usr/bin/python3

import RPi.GPIO as GPIO
from time import sleep

gpio_pin = 18
pause_seconds = 1

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(gpio_pin, GPIO.OUT, initial=GPIO.LOW)

while True:
    GPIO.output(gpio_pin, GPIO.HIGH)
    sleep(pause_seconds)

    GPIO.output(gpio_pin, GPIO.LOW)
    sleep(pause_seconds)
```

Make the script executable, and run it:

```
chmod u+x blinking_led.py

./blinking_led.py
```

Enjoy the blinking light! (Again!)

embedded and iot, dotnet, python