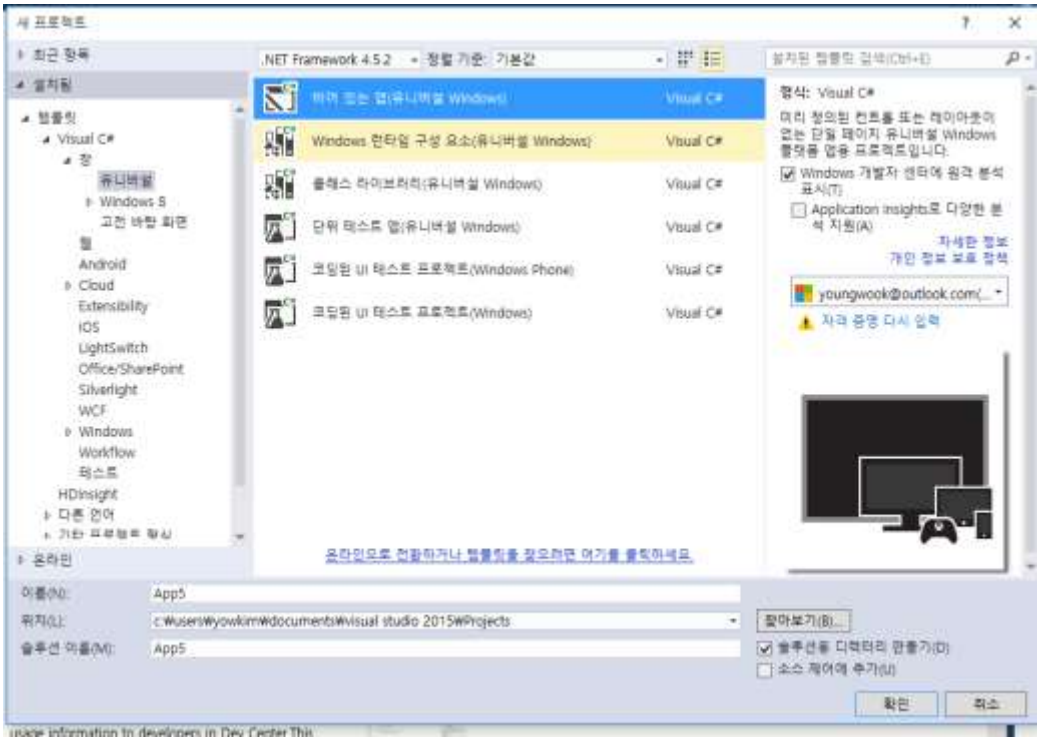
Microsoft IoT Camp

Windows 10 IoT Core 실습

김영욱 Technical Evangelist
부장/ DX / Microsoft

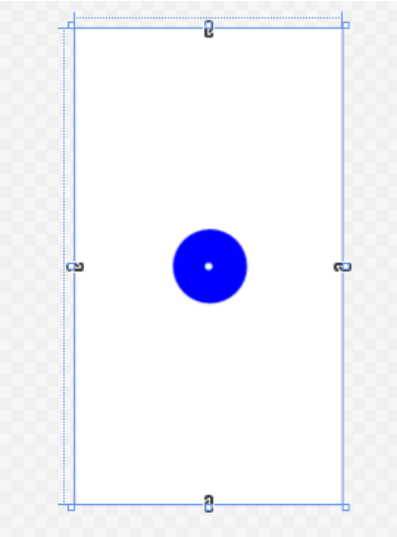
youngwook@outlook.com
Blog: Youngwook.com

먼저 새로운 프로젝트를 생성한다.
C# → 창 → 유니버설 → 비어 있는 앱(유니버설 Windows)



MainPage.xaml에 원을 입력한다.

```
<Ellipse Name="Lamp" Height="100" Width="100" Fill="Blue"/>
```



여기까지 했으면 디자인에 파란색 원이 나타났을 것이다.
MainPage.xaml.cs 파일을 열어서 입력한다.

```
using Windows.UI;
```

멤버 변수로 아래 두 줄을 입력한다.

```
DispatcherTimer timer = new DispatcherTimer();
bool IsLampState = false;
```

생성자에 아래 내용을 입력한다.

```
timer.Interval = TimeSpan.FromMilliseconds(500);
timer.Tick += Timer_Tick;

timer.Start();
```

마지막으로 Timer_Tick( ) 를 작성한다.

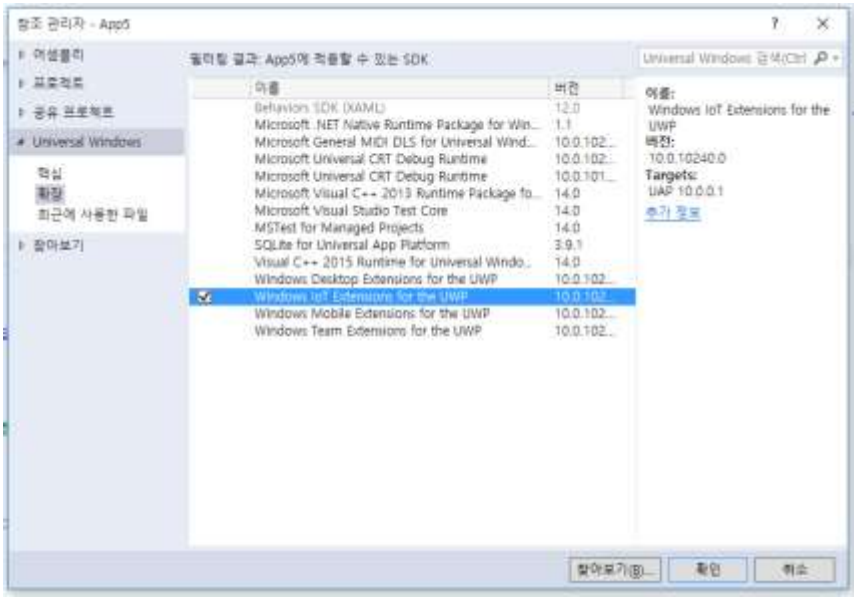
```
private void Timer_Tick(object sender, object e)
{
        if (IsLampState)
        {
                IsLampState = false;
                Lamp.Fill = new SolidColorBrush(Color.FromArgb(255, 0, 0, 255));
        }
        else
        {
                IsLampState = true;
```

```
                Lamp.Fill = new SolidColorBrush(Color.FromArgb(255, 255, 0, 0));
            }
}
```

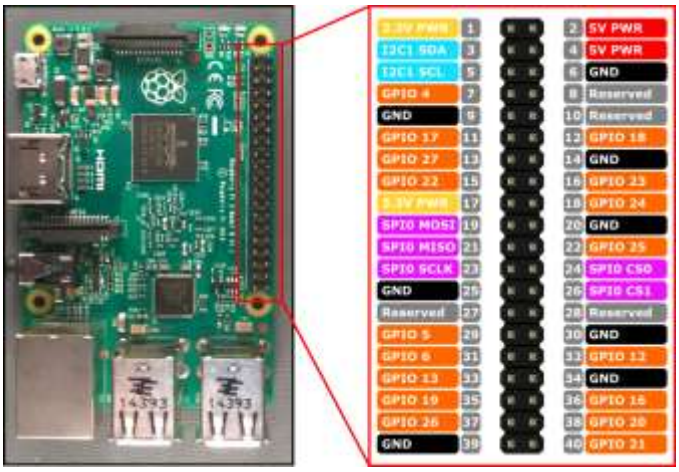
디바이스 제어를 위한 네임스페이스 추가

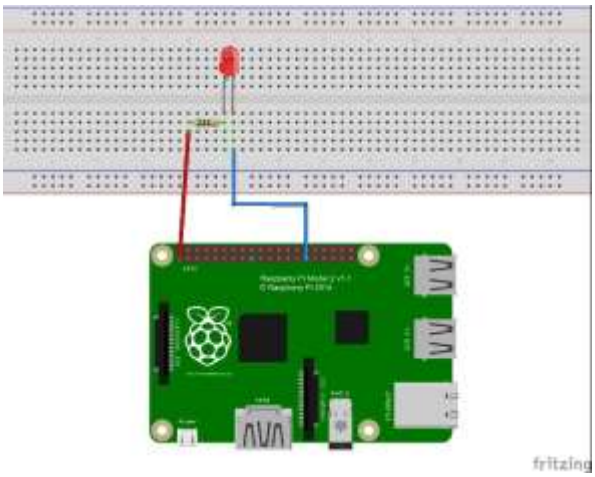
```
using Windows.Devices.Gpio;
```

UWP 확장 모듈을 추가한다.



라즈베리파이의 핀 구성을 참조해서 회로를 만든다.





다시 핀 객체를 맴버로 추가한다.

```
GpioPin Pin5;
```

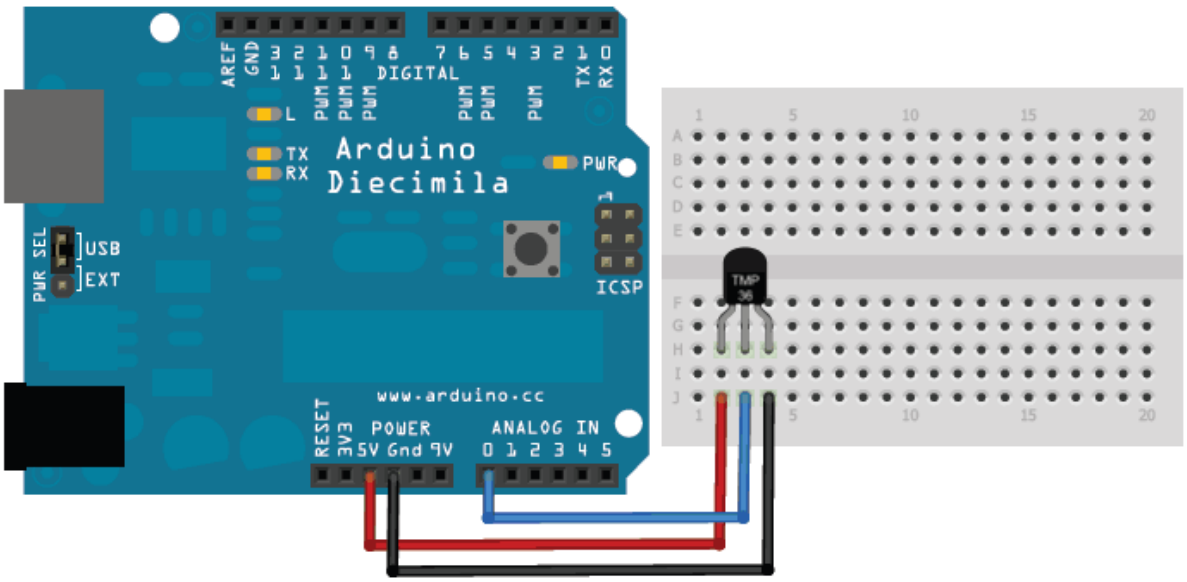
GpioController를 셋팅한다.

```
var gpio = GpioController.GetDefault();
Pin5 = gpio.OpenPin(5);
Pin5.SetDriveMode(GpioPinDriveMode.Output);
```

Timer_Tick( )에 내용을 추가한다.

```
private void Timer_Tick(object sender, object e)
{
            if (IsLampState)
            {
                IsLampState = false;
                Lamp.Fill = new SolidColorBrush(Color.FromArgb(255, 0, 0, 255));
                Pin5.Write(GpioPinValue.Low);
            }
            else
            {
                IsLampState = true;
                Lamp.Fill = new SolidColorBrush(Color.FromArgb(255, 255, 0, 0));
                Pin5.Write(GpioPinValue.High);
            }
}
```

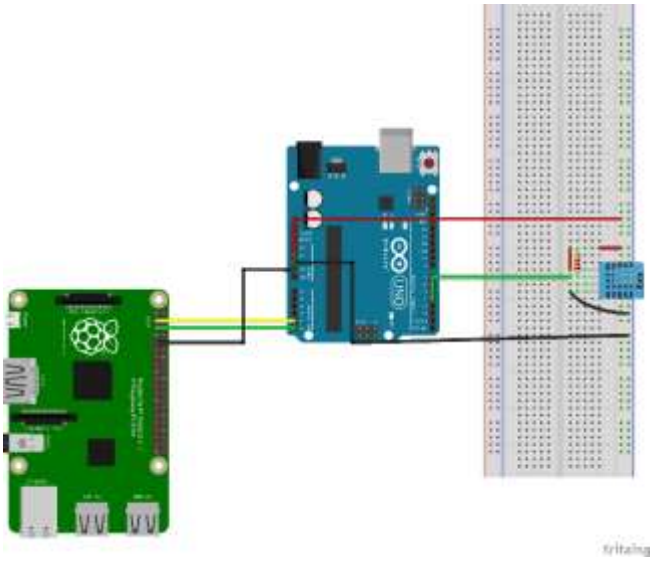
아두이노를 이용해서 온도를 측정하는 코드를 작성한다.




```
#include <Wire.h>
#define SLAVE_ADDRESS 0x40

int sensorPin = 0; //TMP36의 Vout핀과 연결되는 아날로그 핀. 1도당 10mV 변함
char tempera[5];

void setup()
{
   Serial.begin(9600);  //시리얼 콘솔로 결과를 확인하기 위해 PC와 연결 시작
   Wire.begin(SLAVE_ADDRESS);
}

void loop()                     //계속 반복되는 코드
{
  // 온도 센서로부터 Voltage값을 읽어옴
 int reading = analogRead(sensorPin);

 // 읽어들인 값을 Voltage값으로 변환, 3.3V 에 연결했다면 3.3 으로 사용
 float voltage = reading * 5.0;
 voltage /= 1024.0;

 // 온도값을 출력
 float temperatureC = (voltage - 0.5) * 100 ;  // 500mV을 뺀다음 10mV/'C 단위로 바꾸기위해 *100

 Serial.print("S1: == ");
 Serial.println(temperatureC);

 dtostrf(temperatureC, 4, 2, tempera);
 Wire.onRequest(sendData);

 delay(1000);                                  // 1초 간격으로 출력하기 위해 대기
}
```

```
void sendData()
{
  Wire.write(tempera);
}
```

게이트웨이로 아두이노와 통신하는 소스



```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

using Windows.UI;
using Windows.Devices.Gpio;
using Windows.Devices.Enumeration;
using Windows.Devices.I2c;
using System.Diagnostics;
using System.Threading;
//using Microsoft.ServiceBus.Messaging;


// 빈 페이지 항목 템플릿은 http://go.microsoft.com/fwlink/?LinkId=402352&clcid=0x409 에 문서화되어 있습니다.
```

```
namespace App4
{
    /// <summary>
    /// 자체에서 사용하거나 프레임 내에서 탐색할 수 있는 빈 페이지입니다.
    /// </summary>
    public sealed partial class MainPage : Page
    {
        private I2cDevice Device;
        private Timer periodicTimer;

        public MainPage()
        {
            this.InitializeComponent();
            initcomunica();
        }

        private async void initcomunica()
        {

            var settings = new I2cConnectionSettings(0x40); // Arduino address
            settings.BusSpeed = I2cBusSpeed.StandardMode;

            string aqs = I2cDevice.GetDeviceSelector("I2C1");
            var dis = await DeviceInformation.FindAllAsync(aqs);

            Device = await I2cDevice.FromIdAsync(dis[0].Id, settings);

            periodicTimer = new Timer(this.TimerCallback, null, 1000, 1000); // Create a timmer

        }

        private async void TimerCallback(object state)
        {

            byte[] RegAddrBuf = new byte[] { 0x40 };
            byte[] ReadBuf = new byte[5];

            try
            {
                Device.Read(ReadBuf); // read the data
            }
            catch (Exception f)
            {
                Debug.WriteLine(f.Message);
            }

            char[] cArray = System.Text.Encoding.UTF8.GetString(ReadBuf, 0, 5).ToCharArray();  //
Converte  Byte to Char
            String c = new String(cArray);
            Debug.WriteLine(c);
```

```
            try
            {
                var result = await Client.HelloWorldAsync();
            }
            catch(Exception e)
            {
                Debug.WriteLine(e.Message);
            }

        }
    }
}
```