实验13：  
增强Creator IoT WiFire应用程序的能力

北卡罗莱纳州立大学  
电子与计算机工程系

Alexander G. Dean博士  
2016年8月5日，版本1.5



**

*内容*

[概述 4](#_Toc462928440)

[设置 4](#_Toc462928441)

[添加输入 4](#_Toc462928442)

[定义数据结构类型 4](#_Toc462928443)

[创建AnalogInputObject 5](#_Toc462928444)

[复位AnalogInputObject中的统计信息 6](#_Toc462928445)

[测量模拟输入值 6](#_Toc462928446)

[将新的模拟值装入analogInput 6](#_Toc462928447)

[更新AnalogInputObject 7](#_Toc462928448)

[编译和测试代码 7](#_Toc462928449)

# 概述

在本实验练习中，您将通过添加代码以监视作为资源的电位器来增强WiFire板上Creator应用程序的能力。此移动应用程序支持显示电位器的值。

您将修改以下文件：

* firmware\src\ui\_control.c
* firmware\src\app\_client.c
* firmware\src\resources\analog\_input\_object.c

# 设置

1. 您需要按照模块13目录下的文档“安装和使用Creator IoT框架”中的说明进行操作来准备本实验。
2. 然后将MPLABX Lab13\_IoT\_and\_Creator项目归档从本课程模块解压到Harmony WiFIRE SDK安装目录（位于**C:\microchip\harmony\current\apps\**）中。这将创建一个包含此项目的名为**Creator-wifire-lab**的目录。

# 添加输入

我们将添加代码以使用模数转换器读取电位器的值。将使用AnalogInputObject类型的对象，如下所示。该对象跟踪模拟输入的值、最小值、最大值和初始化状态。

|  |
| --- |
| **analogInputObject** |
| Value |
| Min |
| Max |
| SensorType |
| IsInitialised |

在到达对象的存储位置（analogInputObject）之前，模拟数据存储在临时输入缓冲区（被称为analogInput）中。

|  |
| --- |
| **analogInput** |
| Value |

## 定义数据结构类型

1. 定义用于保存输入缓冲区和对象资源的数据结构类型，将以下代码添加到**analog\_input\_object.c**（位于Source Files/app/resources中）中标有“Add type definition code here”（在此处添加类型定义代码）的位置处。

typedef struct // IPSO object: 3202 - AnalogInput

{

AwaFloat Value; // resource 5600

AwaFloat Min; // resource 5601

AwaFloat Max; // resource 5602

char SensorType[SENSORTYPE\_SIZE]; // resource 5751

bool IsInitialised;

} AnalogInputObject;

typedef struct

{

float Value;

} AnalogInput;

1. 接下来定义各种类型的变量。将以下代码添加到**analog\_input\_object.c**中标有“Add variable definitions here”（在此处添加变量定义）的位置处。

static AnalogInputObject analogInputObject[ANALOGINPUT\_INSTANCES];

static AnalogInput analogInput[ANALOGINPUT\_INSTANCES];

## 创建AnalogInputObject

1. **app\_client.c**（位于Source Files/app中）中的函数Client\_Initialise需要有能力创建新对象。将以下代码添加到标有“Add other object creation code here”（在此处添加其他对象创建代码）的位置处。

AnalogInputObject\_Create(\_AwaClient);

1. 函数AnalogInputObject\_Create需要使用Awa客户端定义对象、其资源及其存储位置。将以下代码添加到**analog\_input\_object.c**中标有“Add definition code here”（在此处添加定义代码）的位置处。

AwaStaticClient\_DefineObject(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, "AnalogInput", 0, ANALOGINPUT\_INSTANCES);

AwaStaticClient\_DefineResource(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, IPSO\_ANALOGUE\_INPUT\_CURRENT\_VALUE, "Value", AwaResourceType\_Float, 0, 1, AwaResourceOperations\_ReadOnly);

AwaStaticClient\_SetResourceStorageWithPointer(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, IPSO\_ANALOGUE\_INPUT\_CURRENT\_VALUE, &analogInputObject[0].Value, sizeof(analogInputObject[0].Value), sizeof(analogInputObject[0]));

AwaStaticClient\_DefineResource(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, IPSO\_MIN\_MEASURED\_VALUE, "Min", AwaResourceType\_Float, 0, 1, AwaResourceOperations\_ReadOnly);

AwaStaticClient\_SetResourceStorageWithPointer(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, IPSO\_MIN\_MEASURED\_VALUE, &analogInputObject[0].Min, sizeof(analogInputObject[0].Min), sizeof(analogInputObject[0]));

AwaStaticClient\_DefineResource(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, IPSO\_MAX\_MEASURED\_VALUE, "Max", AwaResourceType\_Float, 0, 1, AwaResourceOperations\_ReadOnly);

AwaStaticClient\_SetResourceStorageWithPointer(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, IPSO\_MAX\_MEASURED\_VALUE, &analogInputObject[0].Max, sizeof(analogInputObject[0].Max), sizeof(analogInputObject[0]));

AwaStaticClient\_DefineResource(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, IPSO\_SENSOR\_TYPE, "SensorType", AwaResourceType\_String, 0, 1, AwaResourceOperations\_ReadOnly);

AwaStaticClient\_SetResourceStorageWithPointer(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, IPSO\_SENSOR\_TYPE, &analogInputObject[0].SensorType, sizeof(analogInputObject[0].SensorType), sizeof(analogInputObject[0]));

1. 初始化存储以及创建对象实例和资源。将以下代码添加到**analog\_input\_object.c**中标有“Add initialization and creation code here”（在此处添加初始化和创建代码）的位置处。

memset(&analogInputObject, 0, sizeof(analogInputObject));

memset(&analogInput, 0, sizeof(analogInput));

int instance;

for (instance = 0; instance < ANALOGINPUT\_INSTANCES; instance++)

{

AwaStaticClient\_CreateObjectInstance(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, instance);

AwaStaticClient\_CreateResource(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, instance, IPSO\_ANALOGUE\_INPUT\_CURRENT\_VALUE);

AwaStaticClient\_CreateResource(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, instance, IPSO\_MIN\_MEASURED\_VALUE);

AwaStaticClient\_CreateResource(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, instance, IPSO\_MAX\_MEASURED\_VALUE);

AwaStaticClient\_CreateResource(awaClient, IPSO\_ANALOGUE\_INPUT\_OBJECT, instance, IPSO\_SENSOR\_TYPE);

strncpy(analogInputObject[instance].SensorType, SENSORTYPE\_POTENTIOMETER, SENSORTYPE\_SIZE);

}

## 复位AnalogInputObject中的统计信息

1. AnalogInputObject保存有关输入数据的统计信息。这些统计信息需要初始化。将以下代码添加到**analog\_input\_object.c**中标有“Add statistics reset code here”（在此处添加统计信息复位代码）的位置处。

int index;

for (index = 0; index < ANALOGINPUT\_INSTANCES; index++)

{

AnalogInputObject \* curAnalogInputObject =

&analogInputObject[index];

if (curAnalogInputObject->IsInitialised)

{

curAnalogInputObject->Min = curAnalogInputObject->Value;

curAnalogInputObject->Max = curAnalogInputObject->Value;

}

}

1. **app\_client.c**中的函数Client\_ResetStatistics需要复位AnalogInputObject的统计信息。将以下代码添加到标有“Call statistics reset code here”（在此处调用统计信息复位代码）的位置处。

AnalogInputObject\_ResetStatistics();

## 测量模拟输入值

1. **ui\_control.c**中的函数UIControl\_pollInputSensors需要使用ADC读取电位器以及更新analogInput数组。通过将以下代码添加到标有“Add analog input code here”（在此处添加模拟输入代码）的位置处来完成上述操作。

AnalogInputObject\_Input(0, AdcDriver\_GetPotentiometerVoltage());

## 将新的模拟值装入analogInput

1. 最新模拟输入值需要存储在analogInput中，以供AnalogInputObject\_Update稍后使用。将以下代码添加到**analog\_input\_object.c**的AnalogInputObject\_Input中标有“Add save input code here”（在此处添加保存输入代码）的位置处。

if (analogInputID < ANALOGINPUT\_INSTANCES)

{

analogInput[analogInputID].Value = value;

}

## 更新AnalogInputObject

1. **app\_client.c**中的函数UpdateResources需要有能力使用数组analogInput中的数据来更新新对象。将以下代码添加到标有“Add other object update code here”（在此处添加其他对象更新代码）的位置处。

AnalogInputObject\_Update(awaClient);

1. 函数AnalogInputObject\_Update用于检查数组analogInput以明确值是否发生更改。如果发生更改，则会将其复制到analogInputObject数组中，更新最小值和最大值，并将资源值更改情况通知Awa客户端。将以下代码添加到**analog\_input\_object.c**中标有“Add update code here”（在此处添加更新代码）的位置处。

int index;

for (index = 0; index < ANALOGINPUT\_INSTANCES; index++)

{

// Check if input changed

AwaFloat value = analogInput[index].Value;

if (analogInputObject[index].Value != value)

{

analogInputObject[index].Value = value;

if (analogInputObject[index].IsInitialised)

{

if (value < analogInputObject[index].Min)

analogInputObject[index].Min = value;

if (value > analogInputObject[index].Max)

analogInputObject[index].Max = value;

}

else

{

analogInputObject[index].IsInitialised = true;

analogInputObject[index].Min = value;

analogInputObject[index].Max = value;

}

AwaStaticClient\_ResourceChanged(awaClient,

IPSO\_ANALOGUE\_INPUT\_OBJECT, index,

IPSO\_ANALOGUE\_INPUT\_CURRENT\_VALUE);

}

}

# 编译和测试代码

1. 编译代码，下载代码，并验证移动应用程序是否正确显示电位器值。您需要按下移动应用程序的重新载入符号（圆形箭头）以从WiFire板获取最新输入值。
2. 将电位器尽可能地逆时针旋转到底。按下移动应用程序上的重新载入符号。应显示0.00 V左右的值。
3. 将电位器尽可能地顺时针旋转到底。按下移动应用程序上的重新载入符号。应显示3.30 V左右的值。
4. 尝试测量按下应用程序上的重新载入符号与观察到新值之间相隔的时间。花费了多长时间？尝试多次重复上述过程。时间是否一致？