Lessons: 00.Arduino-examples

01. Basics

Version: 0.0.3, Date: 2016-10-21

Contents

- AnalogReadSerial
- BareMinimum
- Blink
- BlinkToggle
- BlinkRTTDelay
- BlinkWatchdogDelay
- DigitalReadSerial
- Fade
- VoltageReadSerial

AnalogReadSerial

```
#include “Cosa/AnalogPin.hh”          // Explicit include header files
#include “Cosa/IOStream.hh”
#include “Cosa/UART.hh”

AnalogPin sensor(Board::A0);          // Analog Pin Instance for A0
IOStream ios(&uart);                  // IOStream Instance connected UART

void setup()
{
  uart.begin(9600);                   // Start UART and use 9600 baud
  AnalogPin::powerup();               // Start ADC module
}

void loop()
{
  ios << sensor << endl;              // Print analog pin reading
  delay(100);                         // Delay 100 ms
}
```

AnalogReadSerial Notes

- Explicit include of components used in the sketch.
- Pins are symbols e.g. Board::A0.
- Component initialization in setup().
- IOStream print with operator << and endl.
- Default busy-wait delay().
- Including and initiating the Watchdog will allow low-power sleep mode during delay().

BareMinimum

BareMinimum Notes

- Something missing?
- Cosa has a default setup() and loop() function. The sketch can skip one or both.

Blink

```
#include “Cosa/OutputPin.hh”

OutputPin led(Board::LED);            // Output Pin Instance for LED

void loop()
{
  led.on();                           // Turn LED on and off
  delay(1000);                        // With one second delay
  led.off();
  delay(1000);
}
```

Blink Notes

- This sketch does not need a setup().
- Pin initialization is done by the constructor.
- The OutputPin class allow several methods of accessing:
  - `Pin.on(), Pin.off()`
  - `Pin.set(), Pin.clear()`
  - `Pin.toggle()`
  - `Pin = value`

BlinkRTTDelay

```
#include “Cosa/OutputPin.hh”
#include “Cosa/RTT.hh”

OutputPin led(Board::LED);            // Output Pin Instance for LED

void setup()
{
  RTT::begin();                       // Start Real-Time Timer
}

void loop()
{
  led.on();                           // Turn LED on
  delay(1000);                        // Delay 1000 ms
  led.off();                          // Turn LED off
  delay(1000);                        // Delay 1000 ms
}
```

BlinkToggle

```
#include “Cosa/OutputPin.hh”

OutputPin led(Board::LED);            // Output Pin Instance for LED

void loop()
{
  led.toggle();                       // Change LED state
  delay(1000);                        // Every second
}
```

BlinkWatchdogDelay

```
#include “Cosa/OutputPin.hh”
#include “Cosa/Watchdog.hh”

OutputPin led(Board::LED);            // Output Pin Instance for LED

void setup()
{
  Watchdog::begin();                  // Start Watchdog
}

void loop()
{
  led.on();                           // Turn LED on
  delay(1000);                        // Delay 1000 ms
  led.off();                          // Turn LED off
  delay(1000);                        // Delay 1000 ms
}
```

DigitalReadSerial

```
#include “Cosa/InputPin.hh”           // Explicit include header files
#include “Cosa/IOStream.hh”
#include “Cosa/UART.hh”

InputPin button(Board::D2);           // Input Pin Instance for D2
IOStream ios(&uart);                  // IOStream Instance connected UART

void setup()
{
  uart.begin(9600);                   // Start UART and use 9600 baud
}

void loop()
{
  ios << button << endl;              // Print digital pin reading
  delay(100);                         // Delay 100 ms
}
```

Fade

```
#include "Cosa/PWMPin.hh"             // Explicit include header files

PWMPin led(Board::PWM3);              // PWM Pin Instance on PWM3/D9
int brightness = 0;                   // Current brightness level
int fadeAmount = 5;                   // Brightness adjust amount

void setup()
{
  led.begin();
}

void loop()
{
  led = brightness;
  brightness += fadeAmount;
  if (brightness <= 0 || brightness >= 255)
    fadeAmount = -fadeAmount;
  delay(10);
}
```

VoltageReadSerial

```
#include “Cosa/AnalogPin.hh”          // Explicit include header files
#include “Cosa/IOStream.hh”
#include “Cosa/UART.hh”

AnalogPin sensor(Board::A0);          // Analog Pin Instance for A0
IOStream ios(&uart);                  // IOStream Instance connected UART

void setup()
{
  uart.begin(9600);                   // Start UART and use 9600 baud
  AnalogPin::powerup();               // Start ADC module
}

void loop()
{
  float voltage = sensor * (5.0 / 1023.0);
  ios << voltage << endl;
  delay(100);
}
```