Pronghorn Embedded Toolkit

PET Code examples

A few low level features

- compact 1 Java 8 (<32 MB runtime, -Xmx1m)
- non-blocking lock-free (< 2%CPU on RaspberryPi)
- garbage free runtime (GC pauses are very rare)
- maven builds small uber jar (< 2MB)
  - Using proguard
- BSD Open Source, On GitHub

Build your own projects:
https://github.com/oci-pronghorn/PronghornIoT-Archetype

Review other example projects:
https://github.com/oci-pronghorn/PronghornIoT-Examples

Blinky LED (The ‘hello world’ of IoT)

Demonstrated using the following 3 approaches.

- Using message passing
- Using timer
- Using queue

Build your own projects:
https://github.com/oci-pronghorn/PronghornIoT-Archetype

Review other example projects:
https://github.com/oci-pronghorn/PronghornIoT-Examples

Blink LED - message passing

```
package com.ociweb.iot.project.lightblink;

import static com.ociweb.iot.grove.GroveTwig.LED;

import com.ociweb.iot.maker.CommandChannel;
import com.ociweb.iot.maker.DeviceRuntime;
import com.ociweb.iot.maker.Hardware;
import com.ociweb.iot.maker.IoTSetup;
import com.ociweb.iot.maker.Port;
import static com.ociweb.iot.maker.Port.*;

public class IoTApp implements IoTSetup {

    private static final String TOPIC = "light";
    private static final int PAUSE = 500;
    public static final Port LED_PORT = D4;
```

```
    public static void main( String[] args) {
        DeviceRuntime.run(new IoTApp());
    }

    @Override
    public void declareConnections(Hardware c) {
        c.connect(LED, D4);
    }

    @Override
    public void declareBehavior(DeviceRuntime runtime) {
        /* see next slide */
    }

}
```

Follow me on twitter @NathanTippy

Blink LED - message passing

```
@Override
public void declareBehavior(DeviceRuntime runtime) {
    final CommandChannel blinkerChannel = runtime.newCommandChannel();
    runtime.addPubSubListener((topic,payload)->{

            boolean value = payload.readBoolean();
            blinkerChannel.setValueAndBlock(LED_PORT, value?1:0, PAUSE);
            blinkerChannel.openTopic(TOPIC).writeBoolean(!value).publish();

        }).addSubscription(TOPIC);

    final CommandChannel startupChannel = runtime.newCommandChannel();
    runtime.addStartupListener(()->{
        startupChannel.openTopic(TOPIC).writeBoolean(true).publish();
    });
}
```

Blink LED - timer

```
package com.ociweb.iot.project.lightblink;

import static com.ociweb.iot.grove.GroveTwig.LED;

import com.ociweb.iot.maker.CommandChannel;
import com.ociweb.iot.maker.DeviceRuntime;
import com.ociweb.iot.maker.Hardware;
import com.ociweb.iot.maker.IoTSetup;
import com.ociweb.iot.maker.Port;

import static com.ociweb.iot.maker.Port.*;

public class IoTApp implements IoTSetup {

    private static final int PAUSE = 500;
    public static final Port LED_PORT = D4;
```

```
  public static void main( String[] args) {
      DeviceRuntime.run(new IoTApp());
  }

  @Override
  public void declareConnections(Hardware c) {
      c.connect(LED, D4);
      c.setTriggerRate(PAUSE * 2);
  }

  @Override
  public void declareBehavior(DeviceRuntime runtime) {
      /* see next slide */
  }

}
```

Continuous Integration server on cloudbees:
https://pronghorn.ci.cloudbees.com/view/IoT/

Blink LED - timer (Behavior Declaration)

```
@Override
public void declareBehavior(DeviceRuntime runtime) {
    Final CommandChannel blinkerChannel = runtime.newCommandChannel();

    runtime.addTimeListener((time) -> {
        blinkerChannel.setValueAndBlock(LED_PORT, 1, PAUSE);
        blinkerChannel.setValue(LED_PORT, 0);
    });
}
```

Blink LED - queue

```
package com.ociweb.iot.project.lightblink;

import static com.ociweb.iot.grove.GroveTwig.LED;
import static com.ociweb.iot.maker.Port.*;
import com.ociweb.iot.maker.DeviceRuntime;
import com.ociweb.iot.maker.Hardware;
import com.ociweb.iot.maker.IoTSetup;
import com.ociweb.iot.maker.Port;

public class IoTApp implements IoTSetup {
    public static Port LED_PORT = D4;

    public static void main( String[] args) {
        DeviceRuntime.run(new IoTApp());
    }

    @Override
    public void declareConnections(Hardware c) {
        c.connect(LED, LED_PORT);
        c.setTriggerRate(100);
    }
```

```
 @Override
 public void declareBehavior(DeviceRuntime runtime) {
   runtime.addTimeListener(new BlinkerBehavior(runtime));
 }
}
```

OCI is looking for Java developers

- Embedded, Internet of things
- Cloud, Machine learning

Send resume to:

info@ociweb.com

Blink LED - queue

```
package com.ociweb.iot.project.lightblink;
import com.ociweb.iot.maker.CommandChannel;
import com.ociweb.iot.maker.DeviceRuntime;
import com.ociweb.iot.maker.TimeListener;

public class BlinkerBehavior implements TimeListener {
      private int state = 0;
      private CommandChannel commandChannel;
      private static final int PAUSE = 500;

      public BlinkerBehavior(DeviceRuntime runtime) {
            commandChannel = runtime.newCommandChannel();
      }

      @Override
      public void timeEvent(long arg0) {
            //keep adding commands if more can be accepted
            while (commandChannel.setValueAndBlock(IoTApp.LED_PORT, state, PAUSE)) {
                  state = (1==state ? 0 : 1);
          }
     }
}
```

Features under development

- Support for JUnit tests of maker behavior on simulated hardware
- Event based HTTPs Get/Post
- HTTPs server for Rest, WebSocket, StaticFile
- Secure and Live patch application

Build your own projects:
https://github.com/oci-pronghorn/PronghornIoT-Archetype

Review other example projects:
https://github.com/oci-pronghorn/PronghornIoT-Examples

Looking for open source contributors to help
Send email to: ntippy@ociweb.com