

# Software

# Requirements

- Easy to program
- Flexible
- Timing modification during runtime
- Low power

# Basic concept

- Domain specific language
- Sequence description
- Bytecode generation
  - Very small
- Bytecode interpreter
  - Very fast

```
import "platform:/
resource/
SensorSpecproject/src/
fbs_hw.ssp";

sequence seq_indicator
repeat {
    switch_on Green_LED;
    wait 1 s;
    switch_off Green_LED;
    wait 1 s;
}

sequence
seq_yellow_indicator
repeat{
    switch_on Yellow_LED;
    wait 500 ms;
    switch_off
Yellow_LED;
    wait 500 ms;
}
```

# Language features

## Modules

- Declare Module
  - UART, PIN, OUT, ADC
  - Parameter (for UART)
  - Flag

```
module Bluetooth UART{  
    parameter heartbeatOnly;  
    parameter gsrOnly;  
    parameter sendAll;  
    parameter sendAlive;  
    flag dataReceived;  
}
```

# Language features

## Sequences

- Sequence Types

- Once, repeat, immediate

```
sequence start_all immediate {  
    seq_on seq_indicator;  
    seq_on seq_yellow_indicator;  
}
```

- Sequence Actions

- Switch on/off Module

- Send (UART)/Sample (ADC/Pin)

# Language features

## Sequences

- Sequence operations
  - Sequence on/off
  - Wait (time)
  - Wait (time or flag)

```
sequence seq_sendData repeat {  
    seq_off seq_heartbeat;  
    switch_on Red_LED;  
    switch_on Bluetooth;  
    wait 132 s or dataReceived;  
    send Bluetooth gsrOnly;  
    switch_off Bluetooth;  
    switch_off Red_LED;  
    seq_on seq_heartbeat;  
    wait 185 s;  
}
```

# Bytecode generation

```
sequence seq_yellow_indicator repeat{  
    switch_on Yellow_LED;  
    wait 500 ms;  
    switch_off Yellow_LED;  
    wait 500 ms;  
}
```

```
public static final byte seq_seq_yellow_indicator[]  
    =new byte[]{  
        Sequence.SWITCH_ON_YELLOW_LED,  
        Sequence.WAIT, (byte)0xf3,(byte)0x1, //499 ms  
        Sequence.SWITCH_OFF_YELLOW_LED,  
        Sequence.WAIT, (byte)0xf3,(byte)0x1, //499 ms  
    };
```

# Bytecode generation

```
//Declarations for module: Yellow_LED

/**
 * Switches on the module Yellow_LED.
 * <br>bytecode: (byte)0x13
 */
public static final byte
SWITCH_ON_YELLOW_LED=(byte)0x13;

/**
 * Switches off the module Yellow_LED.
 * <br>bytecode: (byte)0x14
 */
public static final byte
SWITCH_OFF_YELLOW_LED=(byte)0x14;
```



# Bytecode generation

```
sequence seq_indicator repeat {  
  switch_on Green_LED;  
  wait 1 s;  
  switch_off Green_LED;  
  wait 1 s;  
}
```

```
sequence seq_yellow_indicator repeat{  
  switch_on Yellow_LED;  
  wait 500 ms;  
  switch_off Yellow_LED;  
  wait 500 ms;  
}
```

```
sequence start_all immediate {  
  seq_on seq_indicator;  
  seq_on seq_yellow_indicator;  
}
```

00000000	ef 02 00 08 10 04 e7 03
	11 04 e7 03 01 08 13 04
00000010	f3 01 14 04 f3 01
00000016	

# Interpreter generation

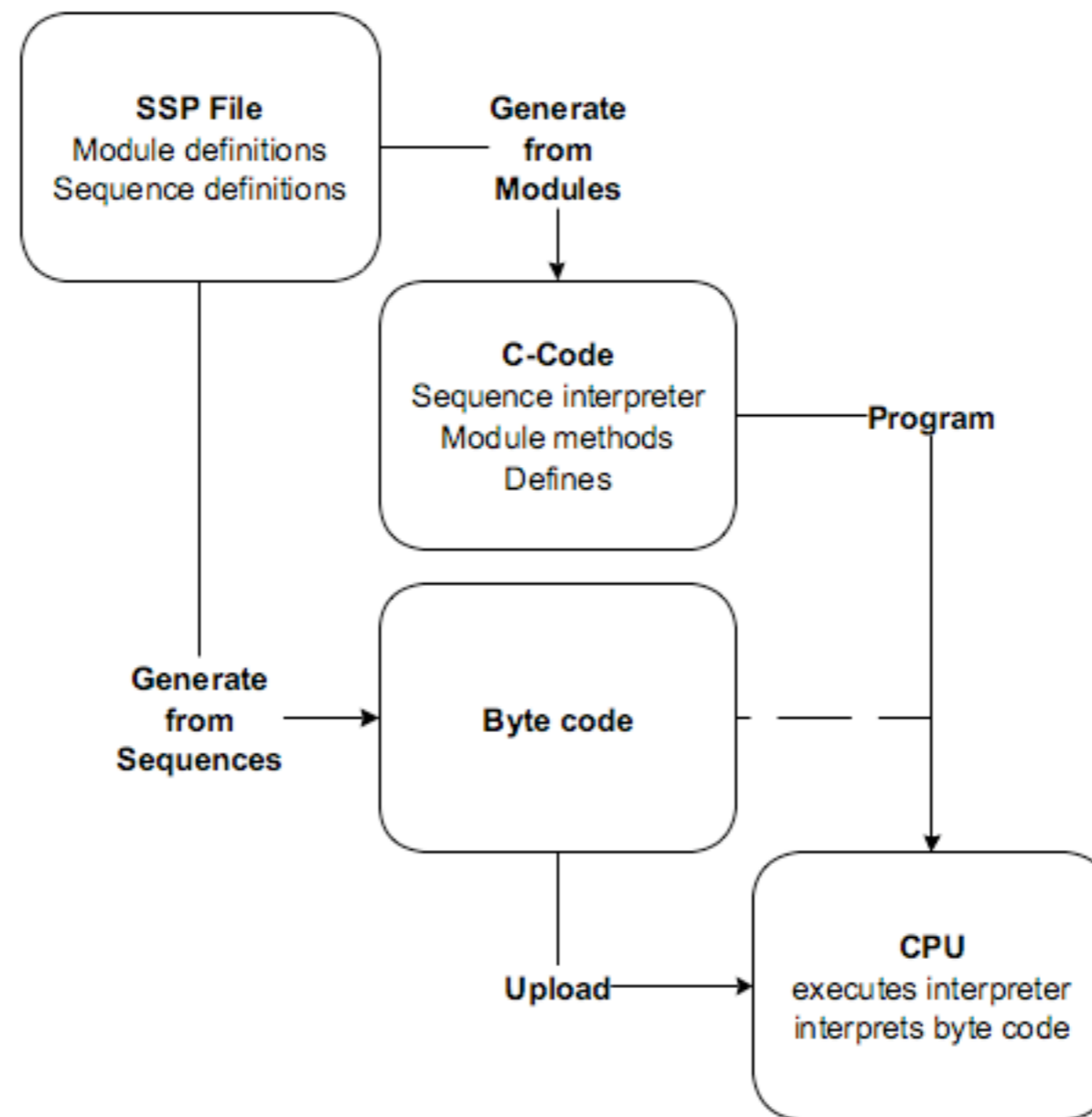
```
//General instructions
#define NOP 0 //Don't do anything
#define NEXT_SEQUENCE 1 //Sequence is done, restart from new next time
#define RESTART 3 //Restart the system
#define WAIT 4 //Wait for some time (the following 2 bytes determine the amount)
#define WAIT_OR 5 //Wait for some time OR a flag. If it expires the sequence is restarted.
#define SEQ_ON 6 //Activate Sequence at pos (next byte)
#define SEQ_OFF 7 //Deactivate Sequence at pos (next byte)

//Declarations for module: Yellow_LED
void init_Yellow_LED(void);
#define SWITCH_ON_YELLOW_LED 19
void switch_on_Yellow_LED(u32);
#define SWITCH_OFF_YELLOW_LED 20
void switch_off_Yellow_LED(u32);
```

# Interpreter generation

```
void evaluateSequence(Sequence *seq, u32 timestamp){
    s8 *currentPos=seq->start+seq->position;
    s8 *end=seq->start+seq->length;
    while (currentPos<end) {
        switch((u8)*currentPos) {
            case NOP:
                break;
            ...
            case SWITCH_ON_YELLOW_LED:
                switch_on_Yellow_LED(timestamp);
                break;
            case SWITCH_OFF_YELLOW_LED:
                switch_off_Yellow_LED(timestamp);
                break;
```

# Software summary



# Realization

- OpenArchitectureWare
- XText
  - Textual domain specific language
  - Automatic generation of
    - Eclipse Editor
    - Code generator

# Eclipse Editor

- Syntax highlighting
- Content assist
- Linking
- Outline