

Follower Code

```
// Define module variables
```

```
TemplateState CurrentState
```

```
uint8_t CurrentCommand = 0
```

```
uint8_t LastCommand
```

```
uint8_t MyPriority
```

```
// Define module functions
```

```
bool InitSPIFollower3(uint8_t Priority):
```

```
    Initialize GPIO and SPI settings
```

```
    Initialize interrupt settings
```

```
    Set CurrentState to InitPStateFollower
```

```
    Create an initialization event ThisEvent with type ES_INIT
```

```
    Post ThisEvent to the service with priority MyPriority
```

```
    Return true if posting successful, otherwise return false
```

```
bool PostSPIFollower3(ES_Event_t ThisEvent):
```

```
    Post ThisEvent to the service with priority MyPriority
```

```
    Return true if posting successful, otherwise return false
```

```
ES_Event_t RunSPIFollower3(ES_Event_t ThisEvent):
```

```
    Initialize ReturnEvent with type ES_NO_EVENT
```

```
    Switch on CurrentState:
```

```
        case InitPStateFollower:
```

```
            If ThisEvent is ES_INIT:
```

```
                Transition to state ReadyToWrite
```

```
                Clear SPI1 buffer
```

```
            Break
```

```
        case ReadyToWrite:
```

```
            Switch on ThisEvent.EventType:
```

Follower Code

Handle various events such as ES_ACORN_DETECTED, ES_IR_DETECTED, etc.:

Send corresponding commands via SPI

Handle ES_EOT event:

Retrieve and handle the command received from the leader

Default:

Do nothing

Break

Default:

Do nothing

Return ReturnEvent

TemplateState_t QuerySPIFollower3(void):

Return CurrentState

void __ISR (_SPI_1_VECTOR, IPL7SOFT)EOT_INTERRUPT(void):

Handle SPI interrupt:

If SPI1RXIF is set:

Read received command from SPI1BUF

If the received command is different from the last command:

Create an ES_Event_t ThisEvent with type ES_EOT and the received command as EventParam

Post ThisEvent to the SPIFollower3 service

Update LastCommand with the received command

Clear SPI1RXIF flag

void PostEvent(uint8_t Command):

Switch on Command:

Handle various commands (e.g., 0x01, 0x02, 0x03):

Create corresponding events and post them to the SPIFollower3 service

Follower Code

```
void __ISR(_INPUT_CAPTURE_5_VECTOR, IPL6SOFT) Acorn_ISR(void):
```

Handle input capture interrupt for Acorn detection:

Increment counter

If RB13 is low:

Disable IC5 interrupt

Create an ES_Event_t ThisEvent with type ES_ACORN_DETECTED

Post ThisEvent to the SPIFollower3 service

Print "!" to indicate Acorn detection

Clear IC5IF flag