Task Scheduler

```
// task definition
typedef struct task {
    unsigned long taskPeriod;
    unsigned long elapsedTime;
    int state;
    int (*TickFct) (int);
} task;

// Global variables for tasks
task tasks[4];
const unsigned char tasksNum = 4;

const unsigned long tasksPeriodGCD = 100;

const unsigned long buttonPeriod = 200;
const unsigned long temperaturePeriod = 500;
const unsigned long ledPeriod = 500;
const unsigned long outputPeriod = 1000;
```

Tasks

```
// button task for when to check button states
 tasks[i].state = BUTTON_FLAG_INIT;
 tasks[i].taskPeriod = buttonPeriod;
 tasks[i].elapsedTime = tasks[i].taskPeriod;
 tasks[i].TickFct = &TickFct_CheckButtonState;
 ++i;

 // temperature task for when to read the temperature
 tasks[i].state = TEMPERATURE_INIT;
 tasks[i].taskPeriod = temperaturePeriod;
 tasks[i].elapsedTime = tasks[i].taskPeriod;
 tasks[i].TickFct = &TickFct_CheckTemperatureState;
 ++i;

 // task for setting the LED states
 tasks[i].state = LED_INIT;
 tasks[i].taskPeriod = ledPeriod;
 tasks[i].elapsedTime = tasks[i].taskPeriod;
 tasks[i].TickFct = &TickFct_CheckLEDState;
 ++i;

 // task for outputting data to server (UART)
 tasks[i].state = OUTPUT_INIT;
 tasks[i].taskPeriod = outputPeriod;
 tasks[i].elapsedTime = tasks[i].taskPeriod;
 tasks[i].TickFct = &TickFct_CheckOutputState;
```

```
// task scheduler interates over tasks and calls the appropriate TickFnc if the elapsed time
// is greater than or equal to each task's period.
unsigned char i;
for (i = 0; i < tasksNum; ++i) {
    if (tasks[i].elapsedTime >= tasks[i].taskPeriod) {
        tasks[i].state = tasks[i].TickFct(tasks[i].state);
        tasks[i].elapsedTime = 0;
    }
    tasks[i].elapsedTime += tasksPeriodGCD;
}
```

Task 1 - Button Task SM Diagram for TickFct_CheckButtonState(int) shown on page 2

```
enum BUTTON_FLAG_States {BUTTON_FLAG_INIT, BUTTON_FLAG_0, BUTTON_FLAG_1};
int TickFct_CheckButtonState(int);
```

Task 2 - Temperature Task SM Diagram for TickFct_CheckTemperatureState(int) shown on page 3

```
enum TEMPERATURE_States {TEMPERATURE_INIT, TEMPERATURE_READ};
int TickFct_CheckTemperatureState(int);
```

Task 3 - LED Task SM Diagram for TickFct_CheckLEDState(int) shown on page 3

```
enum LED_states {LED_INIT, LED_ON, LED_OFF};
int TickFct_CheckLEDState(int);
```

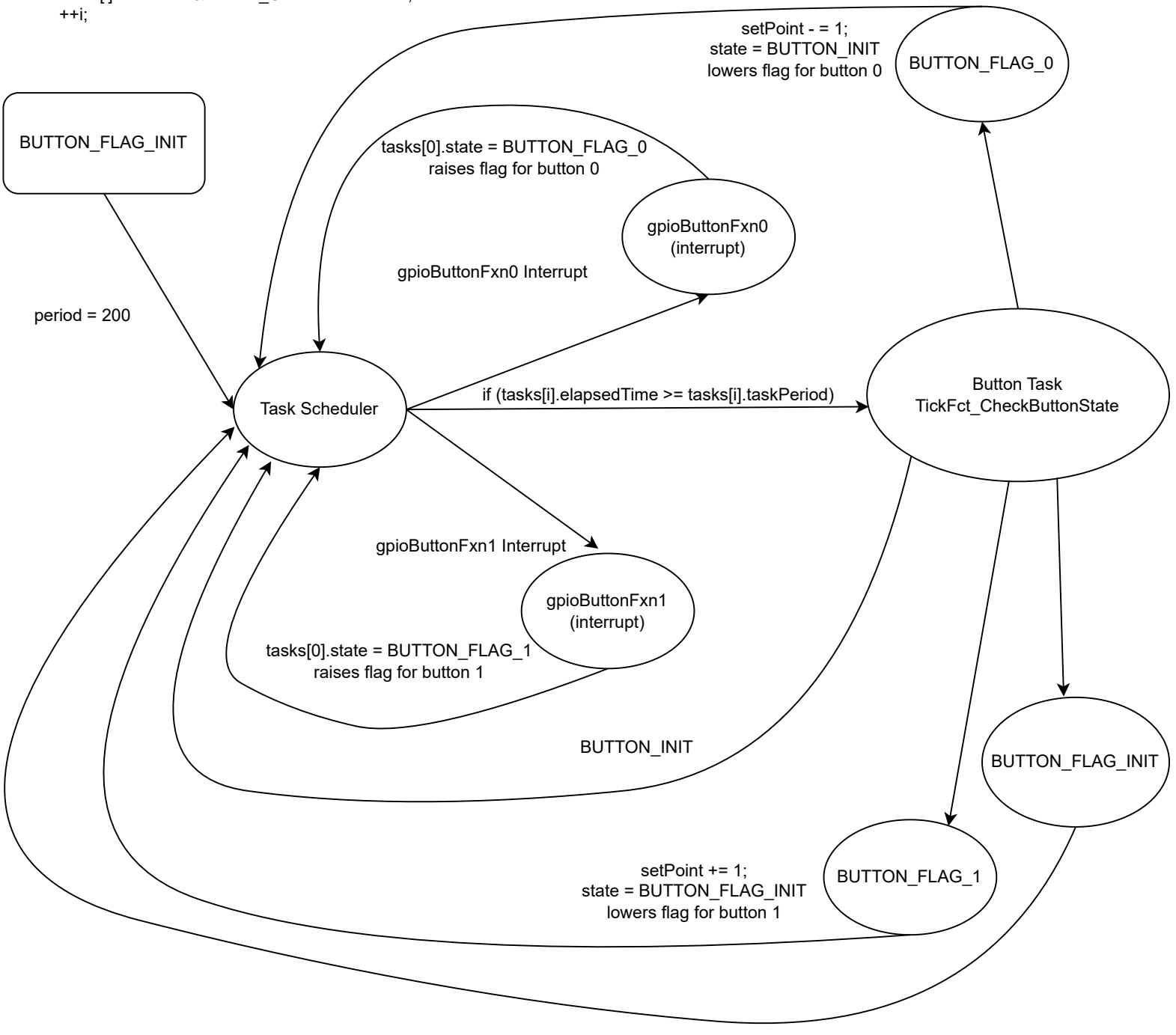
Task 4 - Output Task SM Diagram for TickFct_CheckOutputState(int) shown on page 4

```
enum OUTPUT_States {OUTPUT_INIT, OUTPUT_SEND};
int TickFct_CheckOutputState(int);
```

Task 1 - Button Task SM Diagram for TickFct_CheckButtonState(int)

```
enum BUTTON_FLAG_States {BUTTON_FLAG_INIT, BUTTON_FLAG_0, BUTTON_FLAG_1};
int TickFct_CheckButtonState(int);
```

```
// button task for when to check button states
    tasks[i].state = BUTTON_FLAG_INIT;
    tasks[i].taskPeriod = buttonPeriod;
    tasks[i].elapsedTime = tasks[i].taskPeriod;
    tasks[i].TickFct = &TickFct_CheckButtonState;
    ++i;
```

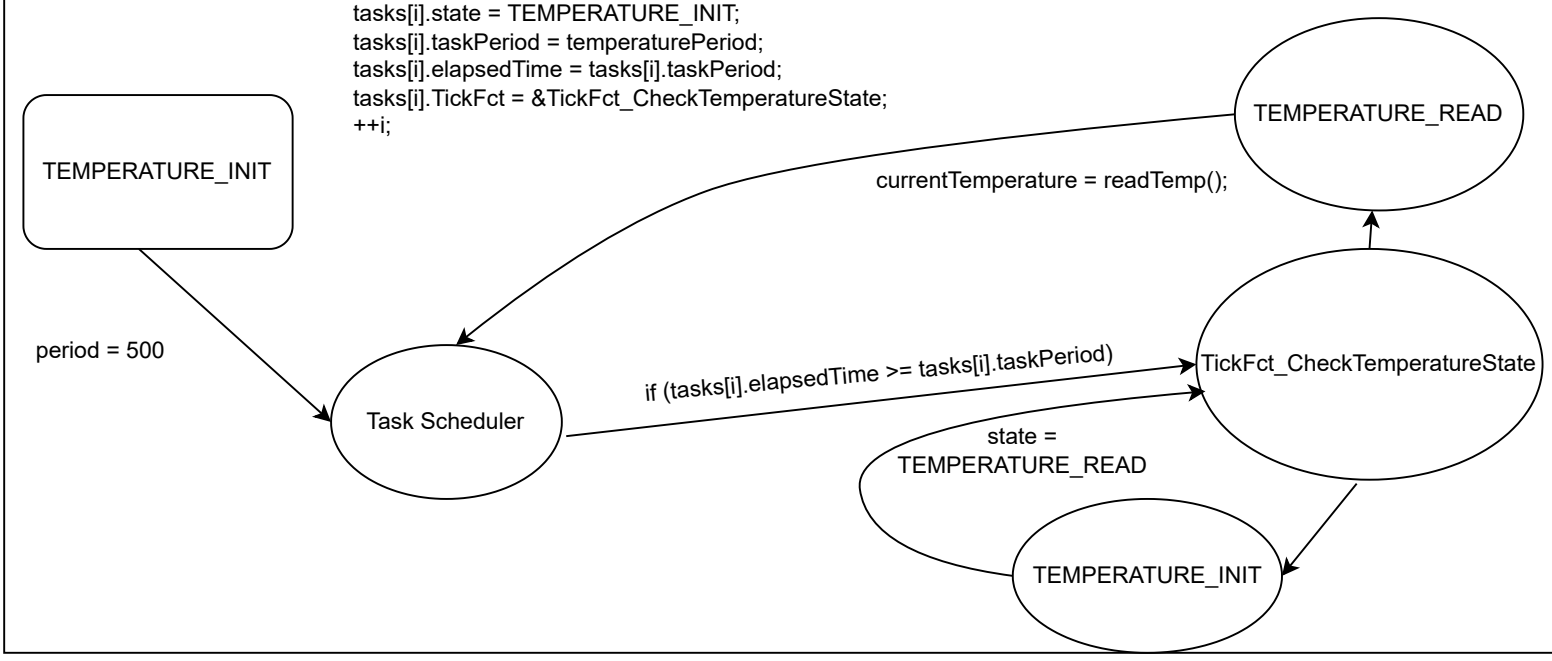


Justin Starr - CS-350 Emerging Systems Architecture & Technologies
SNHU - Department of S.T.E.M - Professor Mike Prasad
February 25, 2024

Task 2 - Temperature Task SM Diagram for TickFct_CheckTemperatureState(int)

```
enum TEMPERATURE_States {TEMPERATURE_INIT, TEMPERATURE_READ};
int TickFct_CheckTemperatureState(int);
```

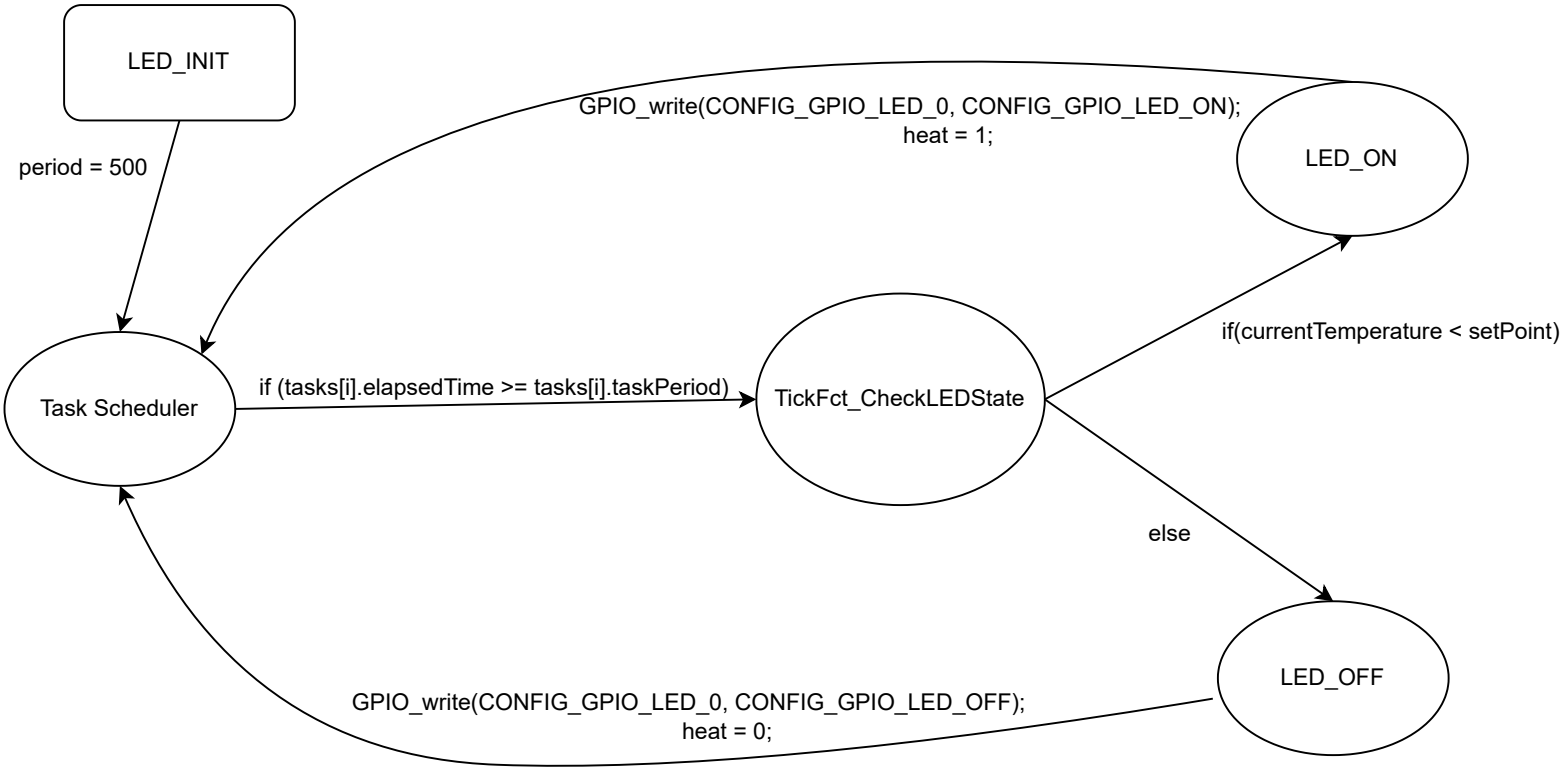
```
// temperature task for when to read the temperature
tasks[i].state = TEMPERATURE_INIT;
tasks[i].taskPeriod = temperaturePeriod;
tasks[i].elapsedTime = tasks[i].taskPeriod;
tasks[i].TickFct = &TickFct_CheckTemperatureState;
++i;
```



Task 3 - LED Task SM Diagram for TickFct_CheckLEDState(int) shown on page 4

```
enum LED_states {LED_INIT, LED_ON, LED_OFF};
int TickFct_CheckLEDState(int);
```

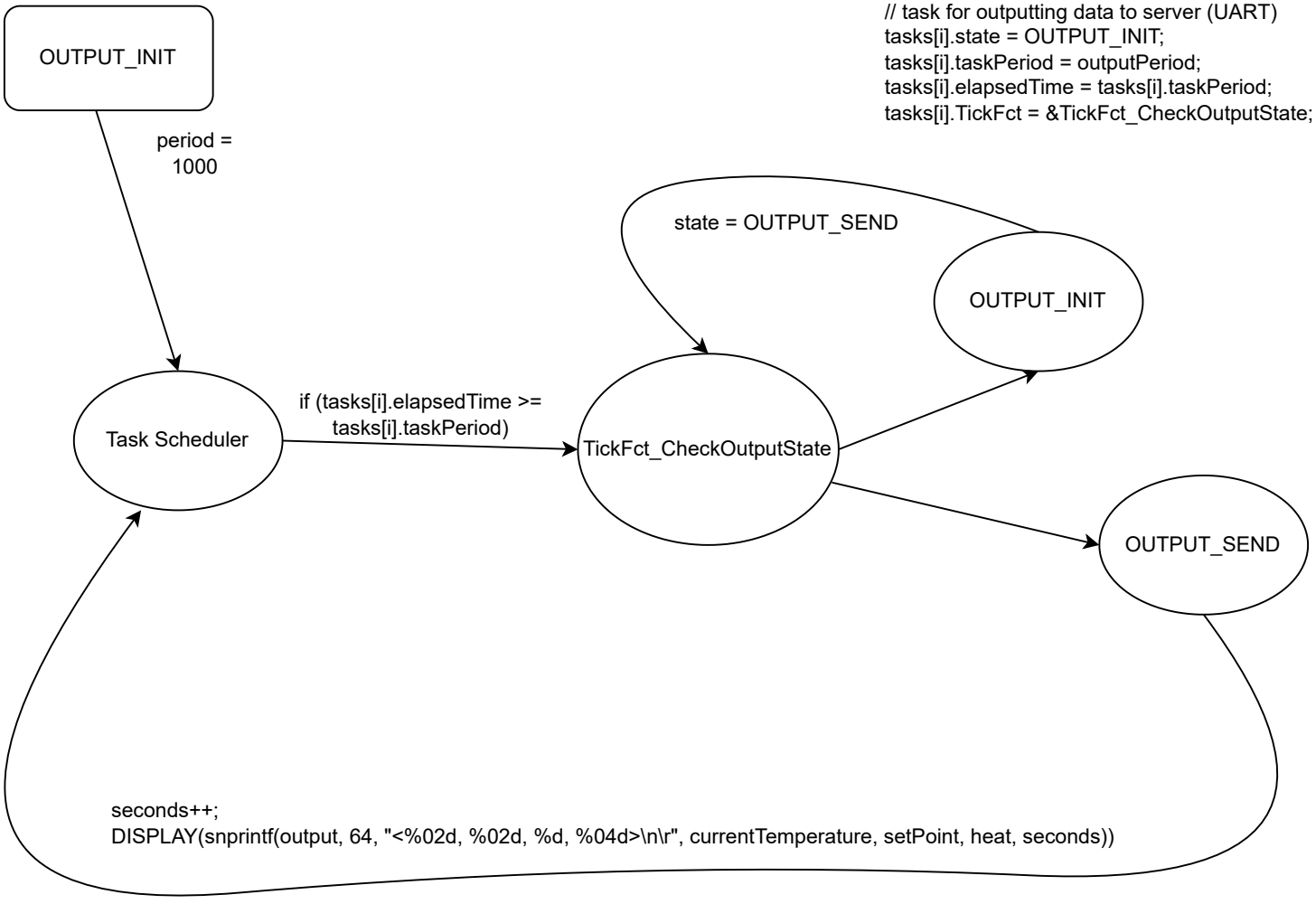
```
// task for setting the LED states
tasks[i].state = LED_INIT;
tasks[i].taskPeriod = ledPeriod;
tasks[i].elapsedTime = tasks[i].taskPeriod;
tasks[i].TickFct = &TickFct_CheckLEDState;
++i;
```



Justin Starr - CS-350 Emerging Systems Architecture & Technologies
SNHU - Department of S.T.E.M - Professor Mike Prasad
February 25, 2024

Task 4 - Output Task SM Diagram for TickFct_CheckOutputState(int) shown on page 4

```
enum OUTPUT_States {OUTPUT_INIT, OUTPUT_SEND};
int TickFct_CheckOutputState(int);
```



Justin Starr - CS-350 Emerging Systems Architecture & Technologies
SNHU - Department of S.T.E.M - Professor Mike Prasad
February 25, 2024