Software Development for Embedded and Realtime Systems

Realtime Building Blocks: Events and Triggers

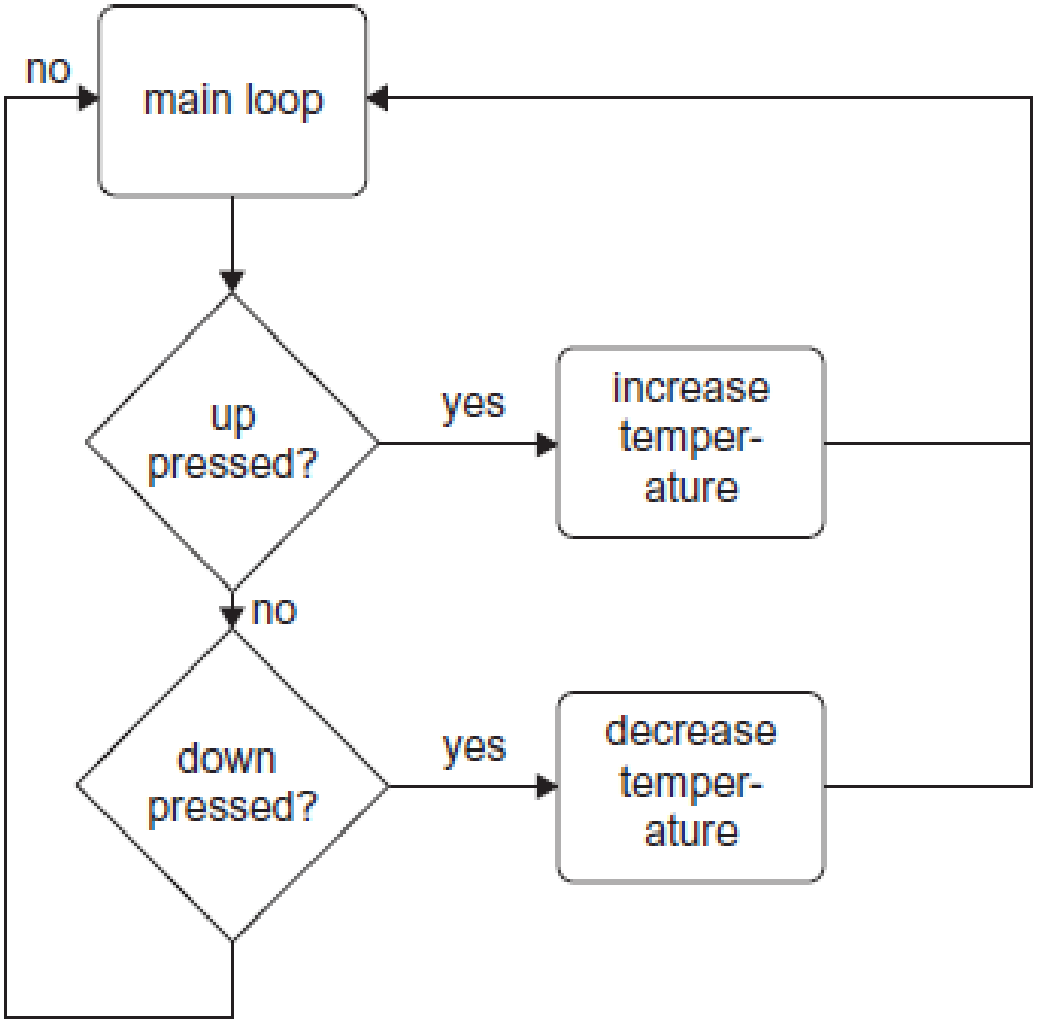
What is an Event?

- An embedded system needs to synchronize with the events from the outside world, and the system itself can create events.
- Examples of events: a button pressed, a sensor reaching a certain value, or the system flags a status to another part of the system.

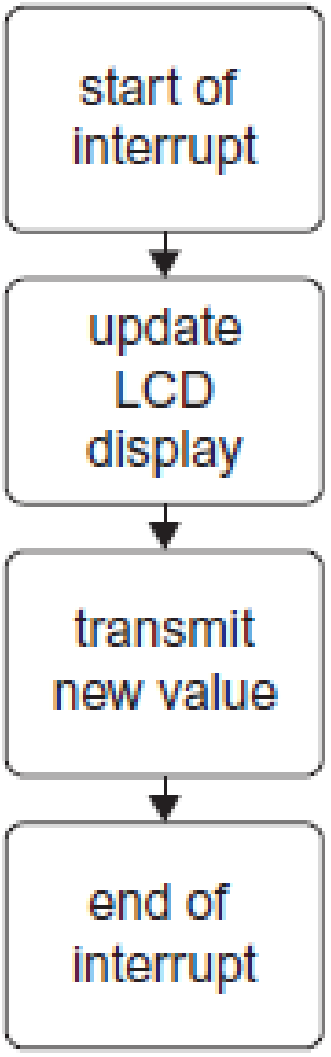
What is a Trigger?

- When an event happens, the system should react in a timely manner.
- Not only has the system to produce the correct result, it has to produce the correct result at the right time.
- A mechanism is needed to perform an action at a guaranteed time. A trigger is used for this purpose.

Example: Air Conditioning System

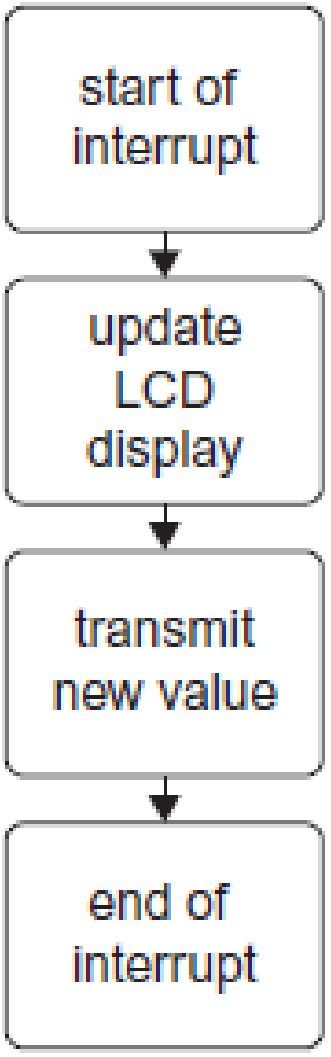


AC System Interrupt Service Routine



- This approach is simple, but has some problems
- updating the display and transmitting the new value might take some time
- all other interrupts might be holding off during the interrupt execution

AC System Interrupt Service Routine



- This approach is simple, but has some problems
- It violates a fundamental design rule for ISRs:

  1. Keep interrupt handlers as small and fast as possible.

  2. Only do things in the interrupt handler which cannot wait.

Event System Design

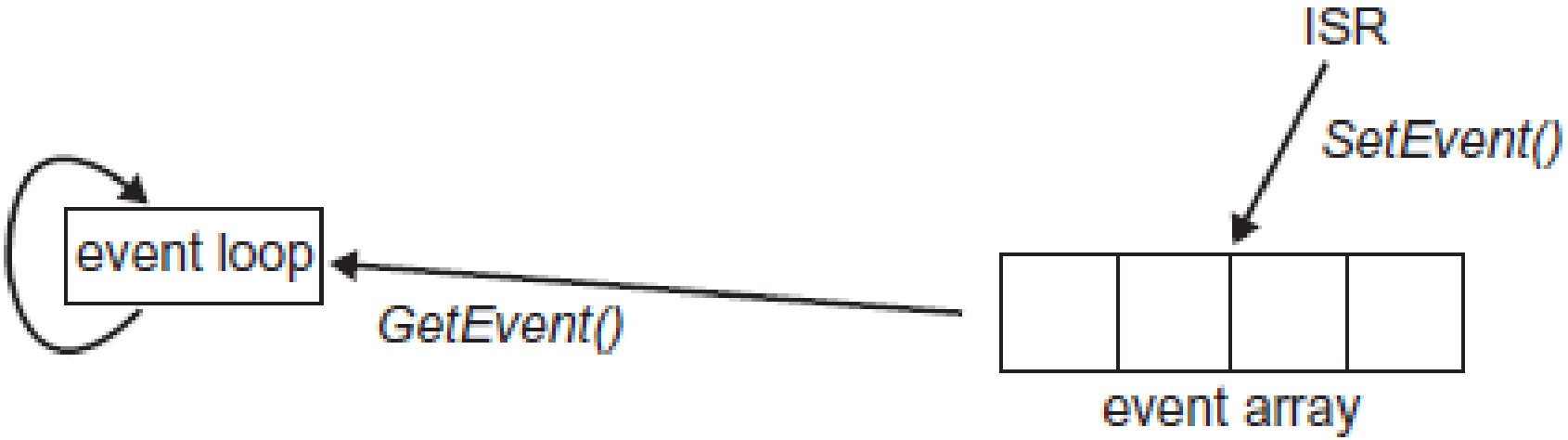


Figure 5.3:
Event ISR system.

Event System Design

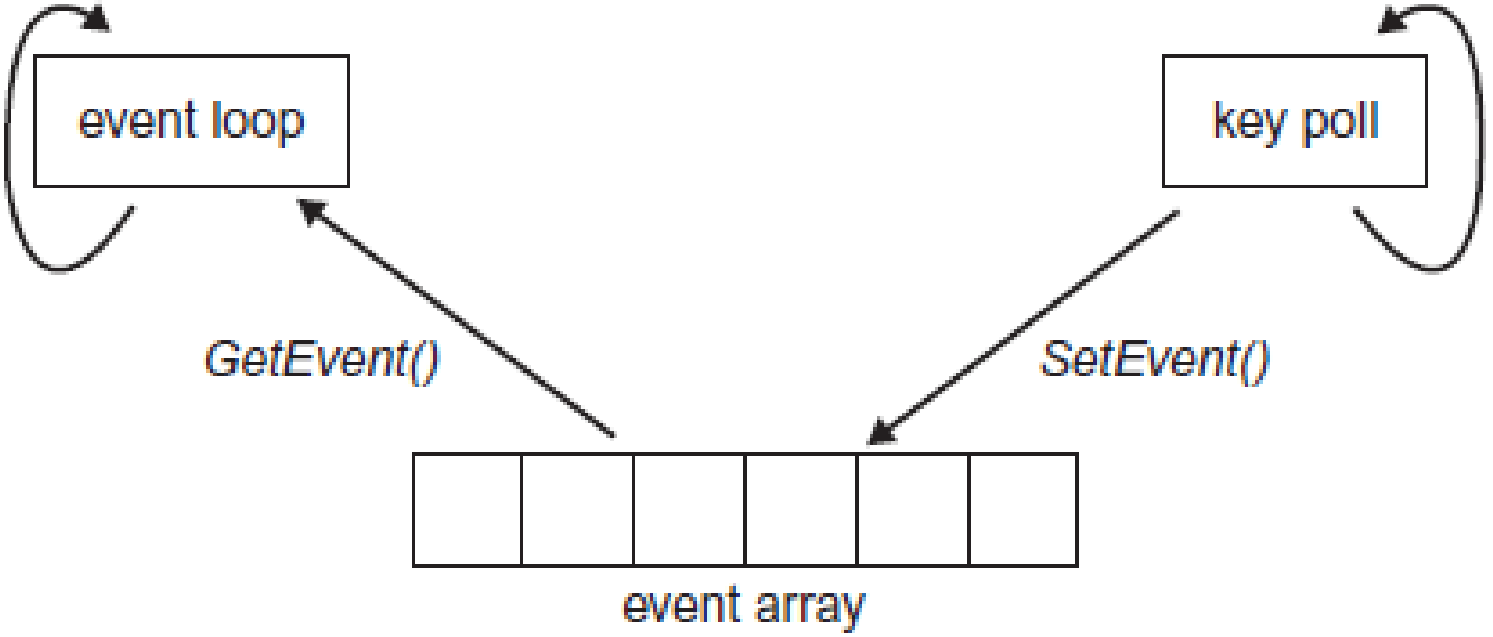


Figure 5.4:
Event polling system.

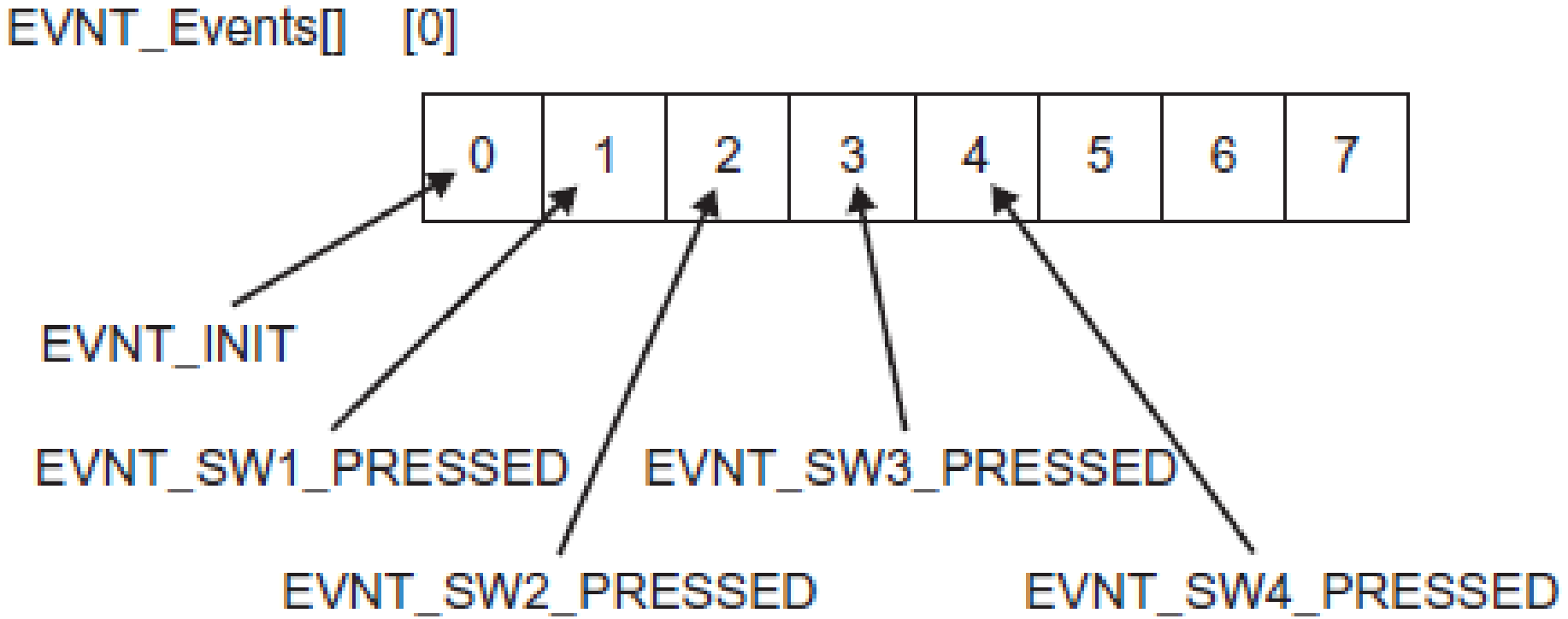
Event Module Design

- Static number of events: number and kind of events are known at compilation time.
- Singularity: an event of any kind can only exist once
- Static memory: a static array of events is used to hold each event handle.
- Event handle: each event owns an event handle or identifier. This event handle is an index into the event array.

Event Types

```
typedef enum {
  EVNT_INIT, /*!< System Initialization Event */
  EVNT_SW1_PRESSED, /*!< SW1 pressed */
  EVNT_SW2_PRESSED, /*!< SW2 pressed */
  EVNT_SW3_PRESSED, /*!< SW3 pressed */
  EVNT_SW4_PRESSED, /*!< SW4 pressed */
  EVNT_NOF_EVENTS, /*!< Must be last one! */
} EVNT_Handle;
```

Event Array


EVNT_Events[] [0]
0
1
2
3
4
5
6
7
EVNT_INIT
EVNT_SW1_PRESSED
EVNT_SW3_PRESSED
EVNT_SW2_PRESSED
EVNT_SW4_PRESSED

Event Methods

- void EVNT_SetEvent(EVNT_Handle event);
- void EVNT_ClearEvent(EVNT_Handle event);
- bool EVNT_GetEvent(EVNT_Handle event);
- void EVNT_HandleEvent(void(*callback)(EVNT_Handle));
- void EVNT_Init(void);

Event Processing – Main Program

```
void main(void) {
EVNT_SetEvent(EVNT_INIT);
for(;;) {
    EVNT_HandleEvent(APP_HandleEvent);
}
}
```

Event Processing – Handling Events

```
void EVNT_HandleEvent(void (*callback)(EVNT_Handle)) {
/* Handle the one with the highest priority.
Zero is the event with the highest priority. */
uint8_t event;
EnterCritical();
/* do a test on every event: */
for (event=0; event<EVNT_NOF_EVENTS; event++) {
  if (GET_EVENT(event)) { /* event present? */
      CLR_EVENT(event); /* clear event */
      break; /* get out of loop */
    }
  }
ExitCritical();
if (event != EVNT_NOF_EVENTS) {
      callback(event);
  }
}
```

Event Processing – Handling Events

```
void APP_HandleEvent(EVNT_Handle event) {
switch(event) {
case EVNT_INIT:
    /* write welcome message */
    LCD_WriteString("System startup. . .");
case EVNT_SW1_PRESSED:
    SND_Beep(300); /* beep for 300 ms */
    /* changes desired temperature */
    ChangeTemperature(1); /* increase temperature */
    SendTemperature(); /* use transceiver */
    break;
case EVNT_SW2_PRESSED:
    SND_Beep(300); /* beep for 300 ms */
    /* changes desired temperature */
    ChangeTemperature(-1); /* decrease temperature */
    SendTemperature(); /* use transceiver */
    break;
} /* switch */
}
```

Event Processing – ISR

```
void interrupt_KeyISR(void) {
ACK_KBI_INTERRUPT(); /* acknowledge interrupt */
if (Key1Pressed()) {
    EVNT_SetEvent(EVNT_SW1_PRESSED);
} else if (Key2Pressed()) {
    EVNT_SetEvent(EVNT_SW2_PRESSED);
}
}
```

Triggers

- Trigger is a way to do something in a time-triggered fashion.
- For example to blink an LED every second, or to turn on an LED 500 ms after a button has been pressed.
- Such a blinking LED is often called a heartbeat.
- An easy way to do this would be to set up a periodic timer which is triggered every 500 ms.

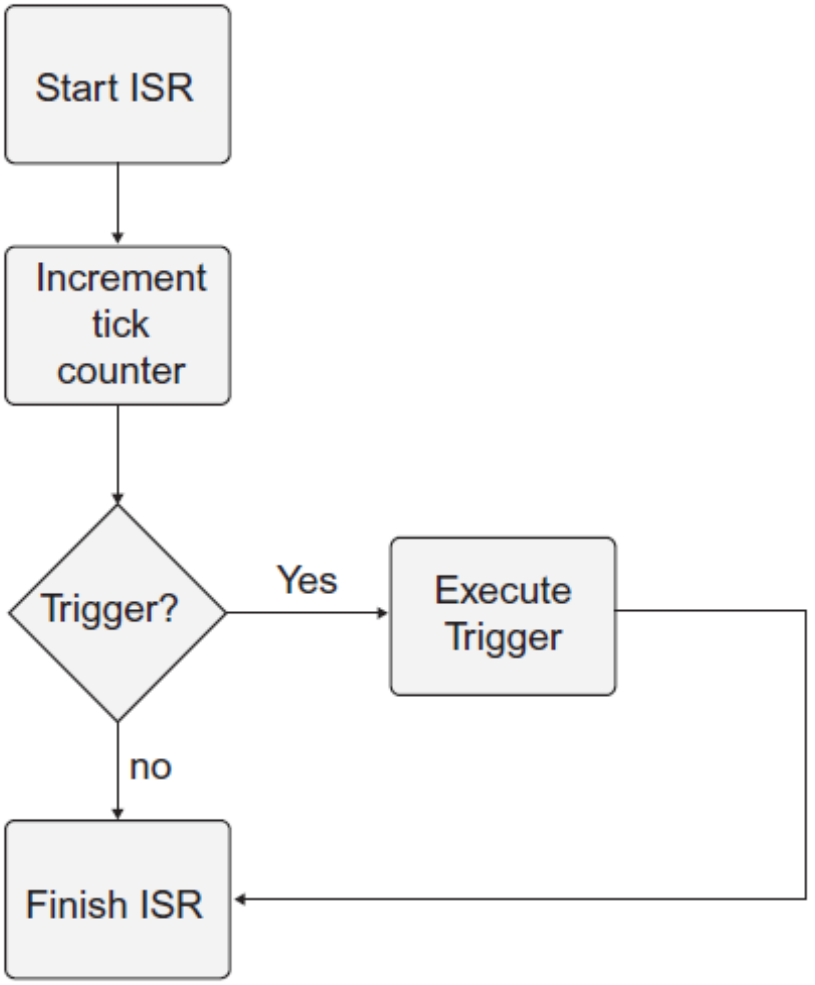
LED Blinking with a Specific Timer

```
interrupt void Timer500 ms(void) {

  LED_Neg(); /* toggle the LED */

}
```

LED Blinking every 500 ms using a 10 ms Timer

```
interrupt void Timer10ms(void) {
 static uint8_t cnt = 0;
 /* blinking LED */
 cnt++; /* increment counter */
 if (cnt = =500/10) { /* 500 ms reached */
  LED_Neg(); /* toggle the LED */
  cnt = 0; /* restart counter */
 }
 /* other things to do every 10 ms follows here...*/
}
```

LED Blinking with a Specific Timer



References

- **Chapter 5**: Oshana, Robert, ed. **Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications.** Newnes, 2013.