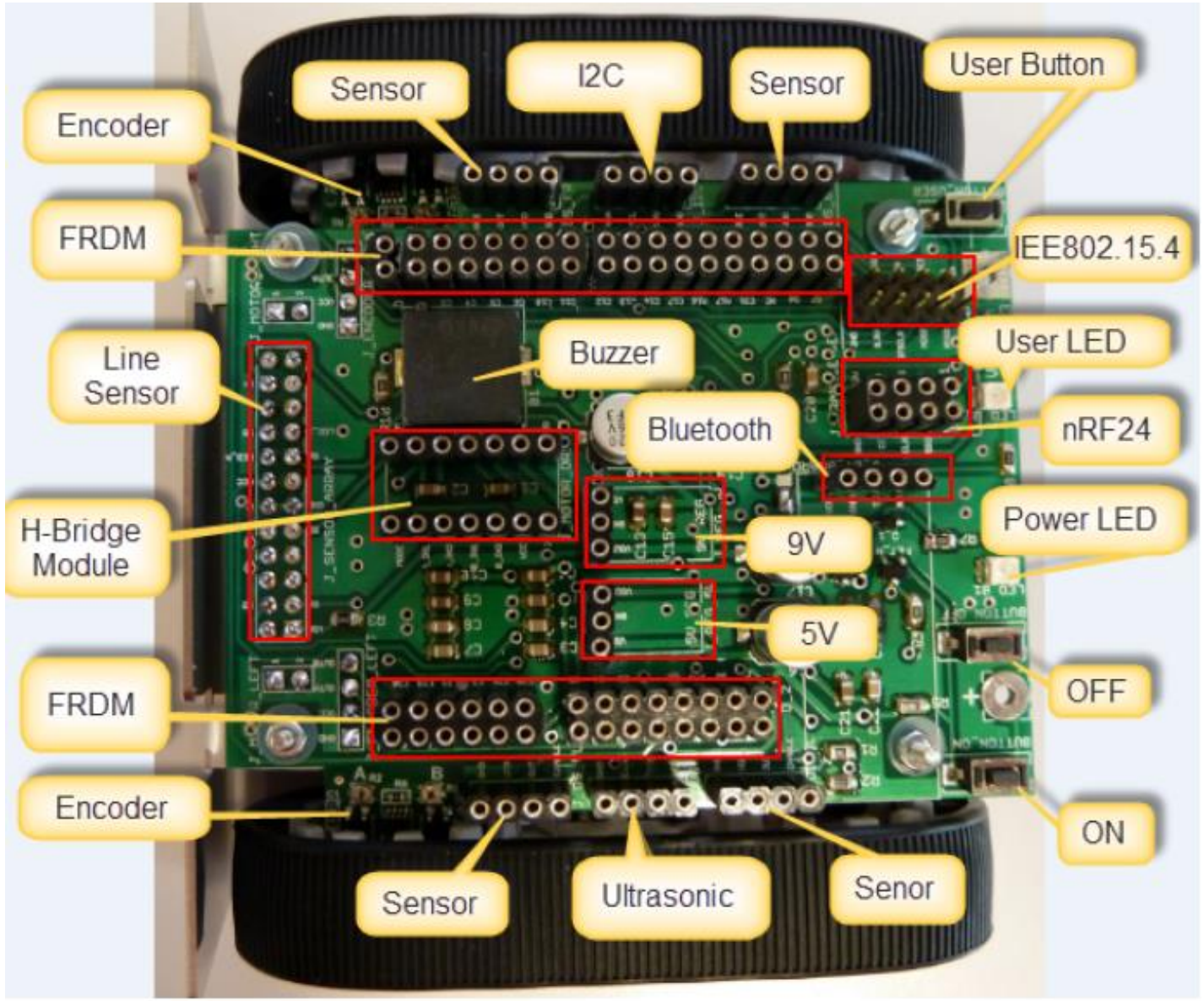
Recap
Monday
10.03.2014

Roman Brun, Marco Fischlin

Table of Contents

Sumo Component Overview



What’s an S19 File?

- Text files with code

```
S00F000068656C6C6F202020202000003C
S11F00007C0802A6900100049421FFF07C6C1B787C8C23783C600000386300002 6
S11F001C4BFFFFE5398000007D83637880010014382100107C0803A64E800020E9
S111003848656C6C6F20776F726C642E0A0042
S5030003F9
S9030000FC
```

Source: Wikipedia

- Start code
- Record type (Datensatztyp)
- Byte count
- Adresse
- Daten
- Checksum (Prüfsumme)

S19 Record types

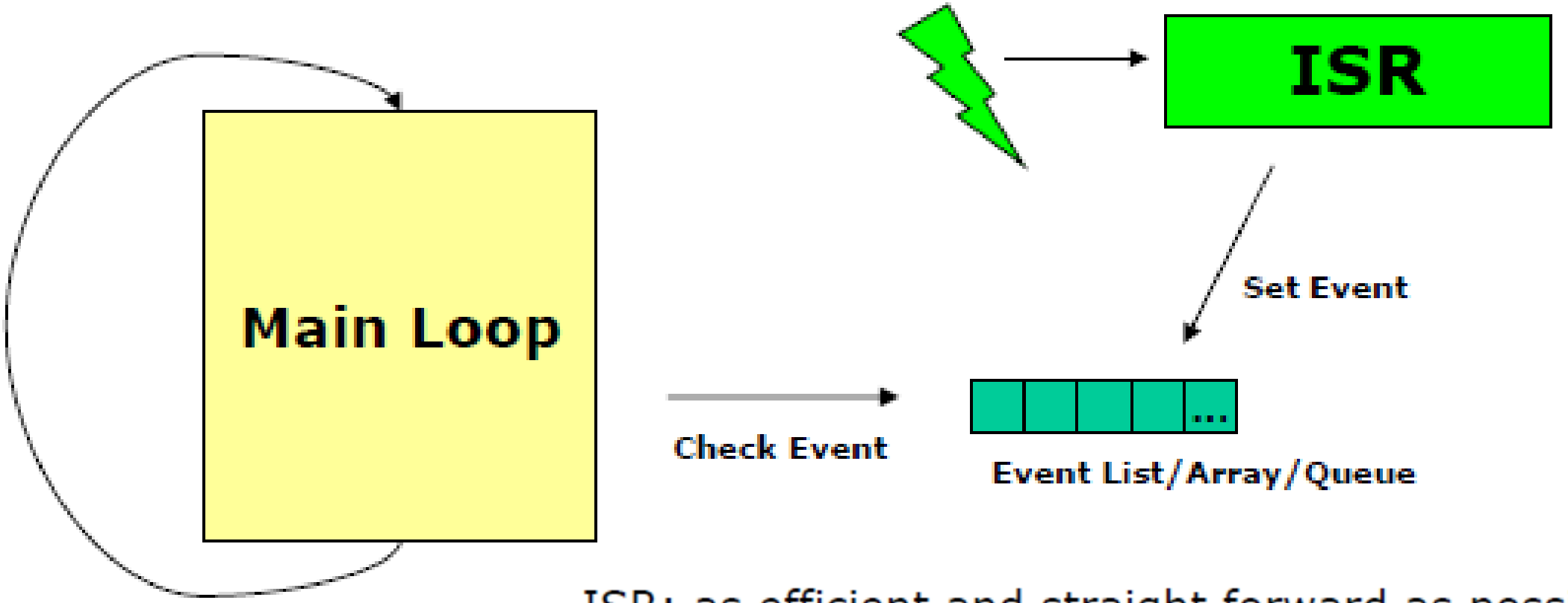
Record	Beschreibung	Adressbytes	Datenfeld
S0	*Block header* (Block Vorspann)	2	Ja
S1	Datenreihe	2	Ja
S2	Datenreihe	3	Ja
S3	Datenreihe	4	Ja
S5	*Record count* (Datensatzanzahl)	2	Nein
S7	*End of block* (Blockende)	4	Nein
S8	*End of block* (Blockende)	3	Nein
S9	*End of block* (Blockende)	2	Nein

Events

- Synchronous Events
  - Timer interrupt
  - Periodic Task output
- Asynchronous Events
  - Button pressed
  - Transceiver packet received

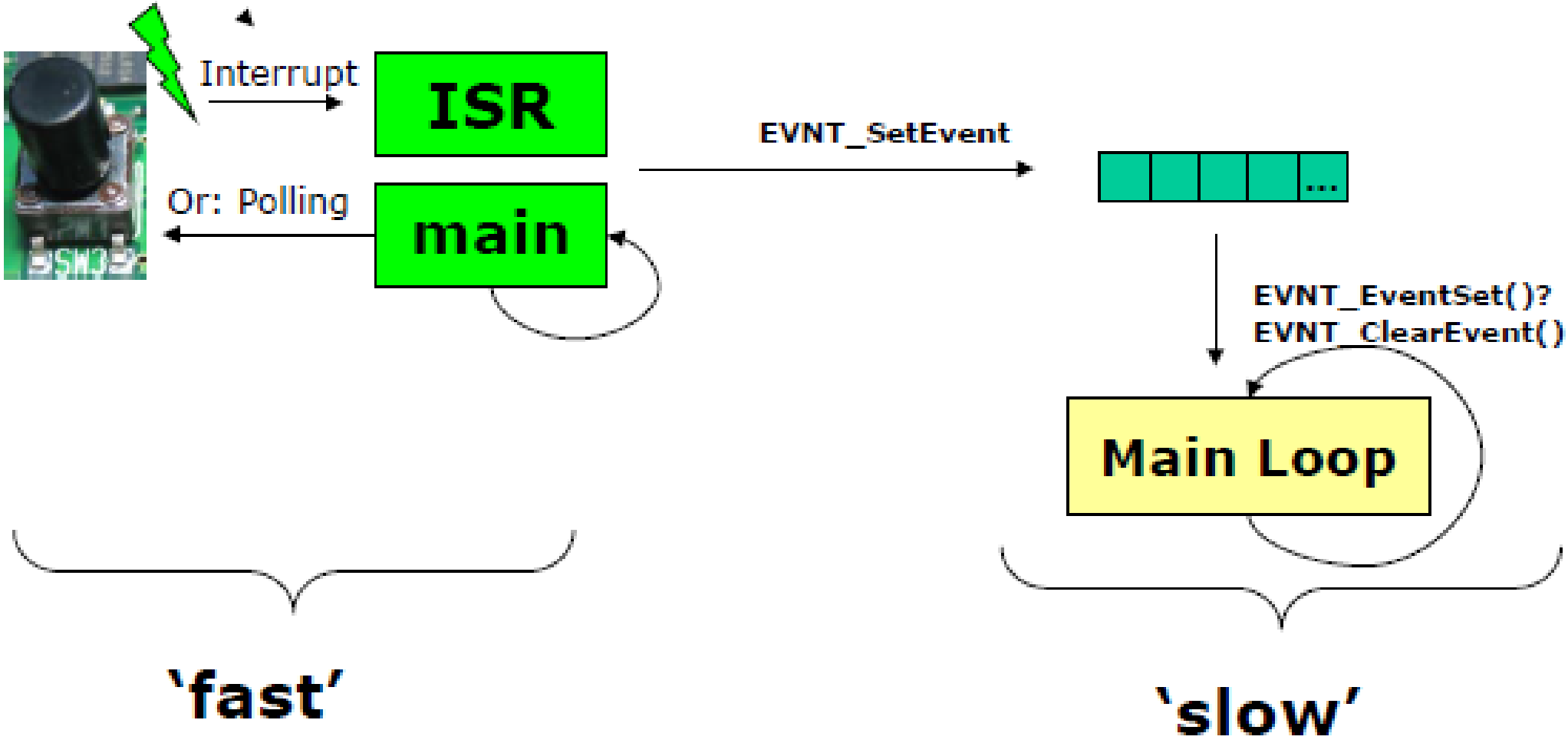
- Need Infrastructure
  - Set/clear/chek if event happend
- Implementation
  - RTOS
  - Flags
  - Possible implementation
    - Queue,List,**Array**

Interrupt Execution Speed



- ISR: as efficient and straight forward as possible
- Possible approach: Event Loop/Handler
- Main loop does the ‘heavy’ workload
- Interrupt Service Routines: Create/Set events (flags)

Decoupling Event Processing



Storing Events

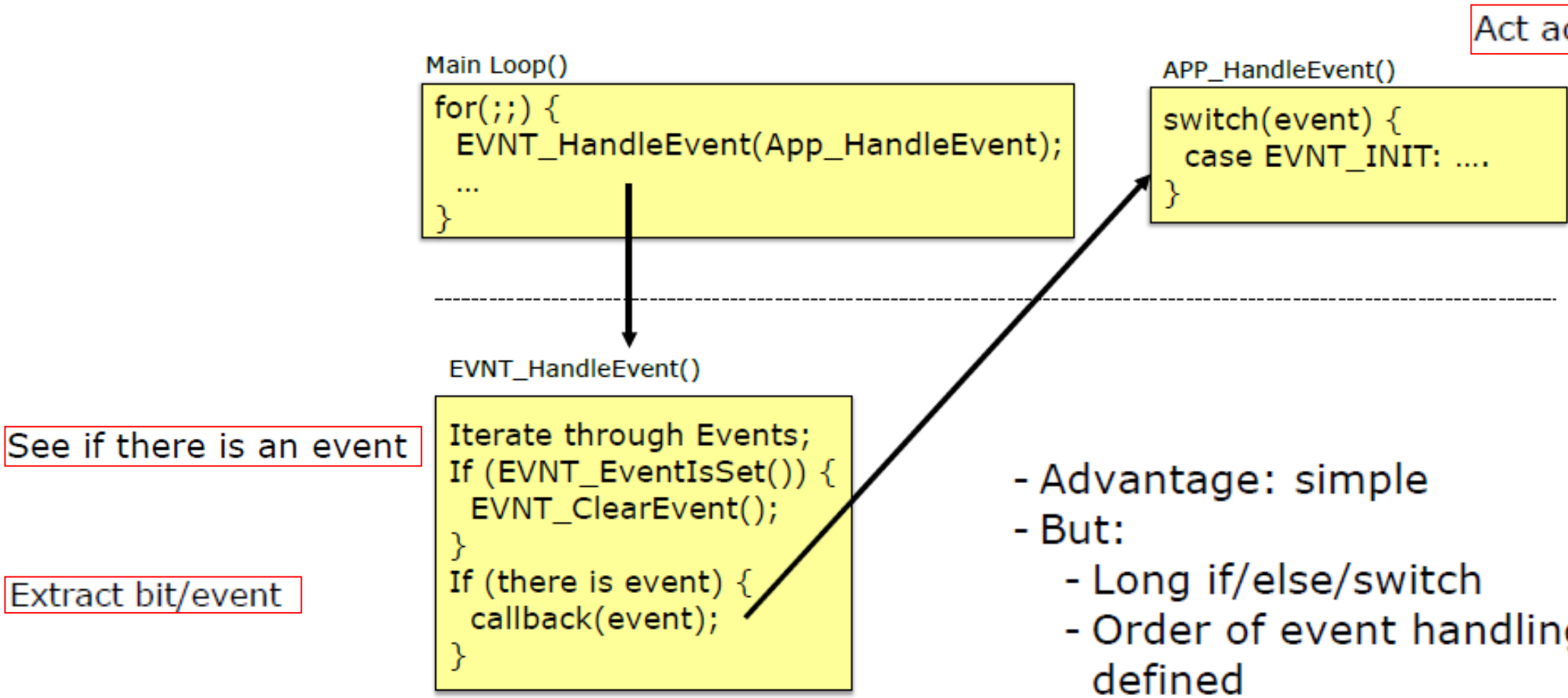
```
typedef enum {
  EVNT_INIT,              /*!< System Initialization Event */
  EVNT_SW1_PRESSED,       /*!< SW1 pressed */
  EVNT_SW2_PRESSED,       /*!< SW2 pressed */
  EVNT_SW3_PRESSED,       /*!< SW3 pressed */
  …
  EVNT_NOF_EVENTS         /*!< Must be last one! */
} EVNT_Handle;
```

Numbering can be priority!

Sentinel at the end
- Gives Number of Event items
- Used for bit array size definition

- Idea
  - Using as few memory as possible
  - Using event 'flags'
  - → mapping from ‘numbers’ to bits/flags

Handling Events from Main Loop



- Advantage: simple
- But:
  - Long if/else/switch
  - Order of event handling needs to be defined
  - Need to protect against concurrent access!

Event Interface

```
void EVNT_SetEvent(EVNT_Handle event);

void EVNT_ClearEvent(EVNT_Handle event);

bool EVNT_EventIsSet(EVNT_Handle event);

void EVNT_HandleEvent(
       void (*callback)(EVNT_Handle));

void EVNT_Init(void);

void EVNT_Deinit(void);
```

Thanks for your attention!

Questions

What is an S19 file?

- It is a file with code.

What is the meaning of S1 in a S19 file?

- It means, that a stream of data follow with a two byte adress.

What kind of events does exist?

- Synchronous Events
  - Timer interrupt
  - Periodic Task output
- Asynchronous Events
  - Button pressed
  - Transceiver packet received

What is the sentinel at the end of the enum type EVNT_Handle for?

- EVNT_NOF_EVENTS

Sentinel at the end
- Gives Number of Event items
- Used for bit array size definition

What are the advantages and disandvantages of handling events from the main loop?

- Advantage: simple
- Disadvantages:
  - Long if/else/switch (in APP_HandleEvent())
  - Order of event handling needs to be defined
  - Need to protect against concurrent access!