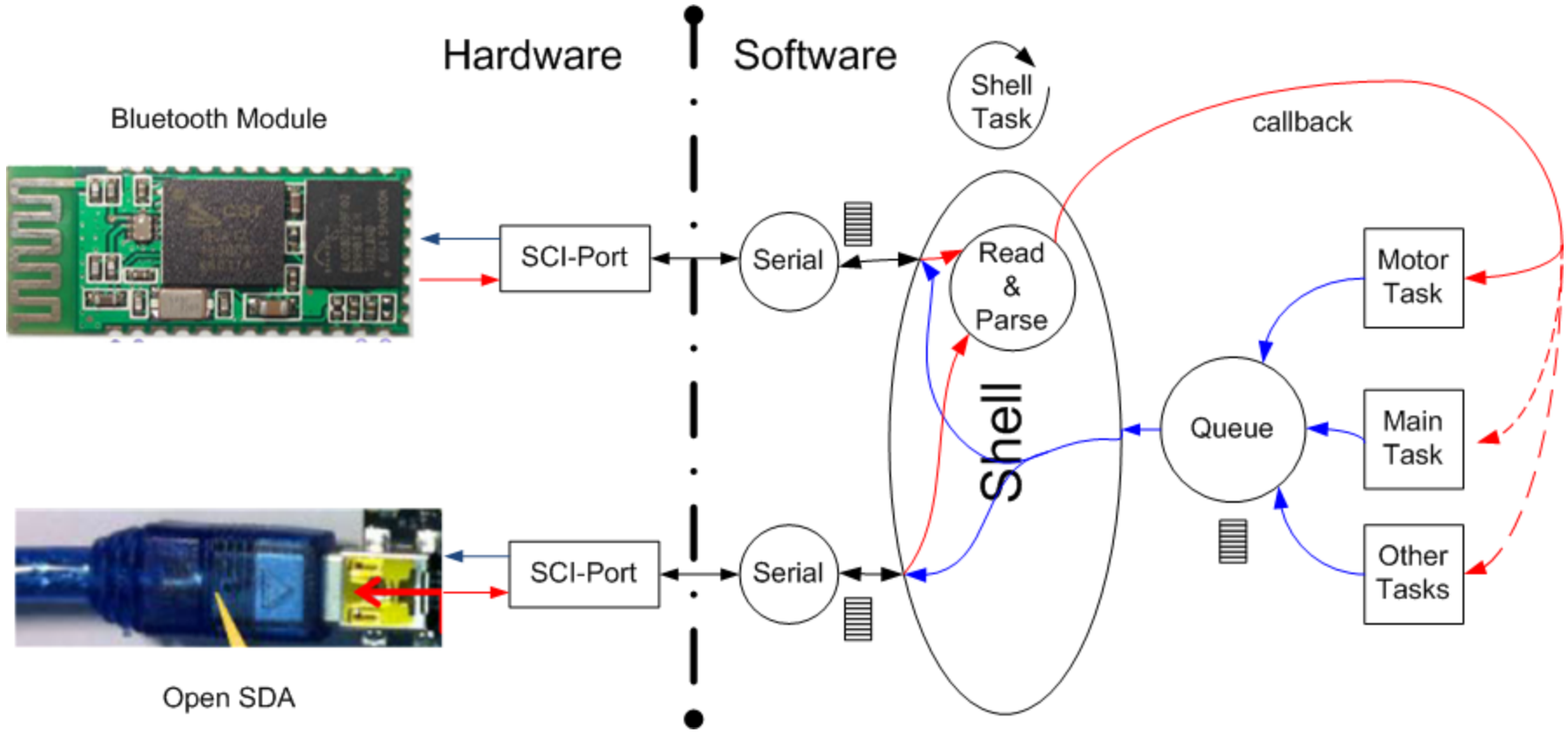
Shell,
USB and UART,
Memory and Queue

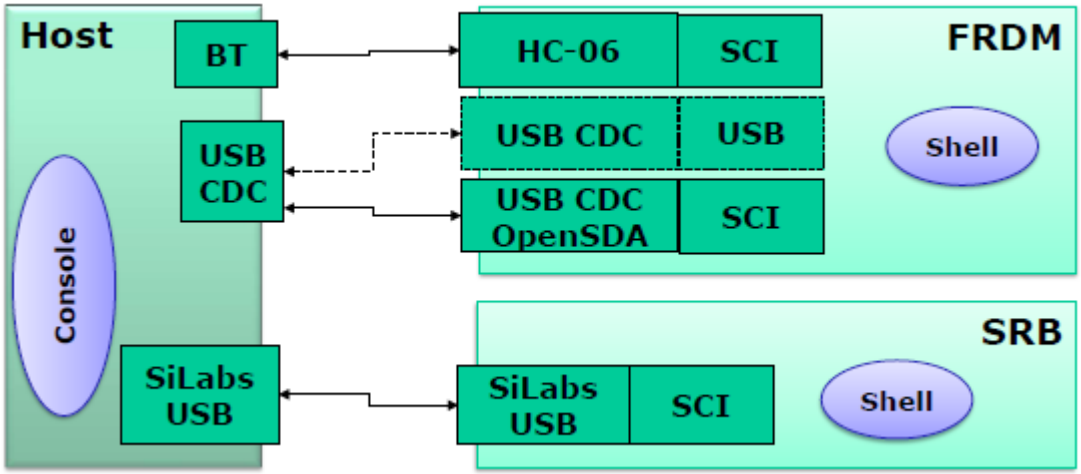
Andreas Walker,

Ruedi Herger,

Reto Müller


Hardware
Software
Bluetooth Module
Shell Task
callback
SCI-Port
Serial
Read & Parse
Motor Task
Shell
Queue
Main Task
SCI-Port
Serial
Other Tasks
Open SDA

Ways to communicate



- Bluetooth to a SCI port of the MCU
- USB bridge to a SCI port of the MCU
- (USB directly to a USB port of the MCU)

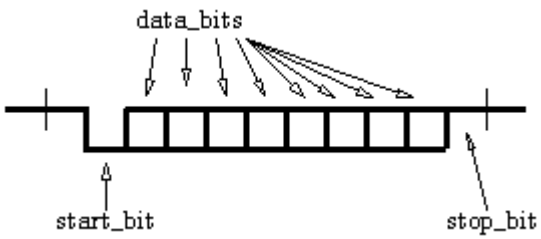
SCI Serial COM Interface (UART)

Wires:

- Rx, receive line
- Tx, send line
- Additionally: flow control lines (not at the FRDM)

Protocol properties:

- Asynchron, serial (no clock)
- Fixed baudrate



Processor expert component: AsynchroSerial
-> Used by the Shell and Bluetooth component

USB

Wires:

- 2 power supply lines
- 2 data lines (differential signal, noise resistant)

Device types:

- Host, device, OTG (On-The-Go)

Device classes:

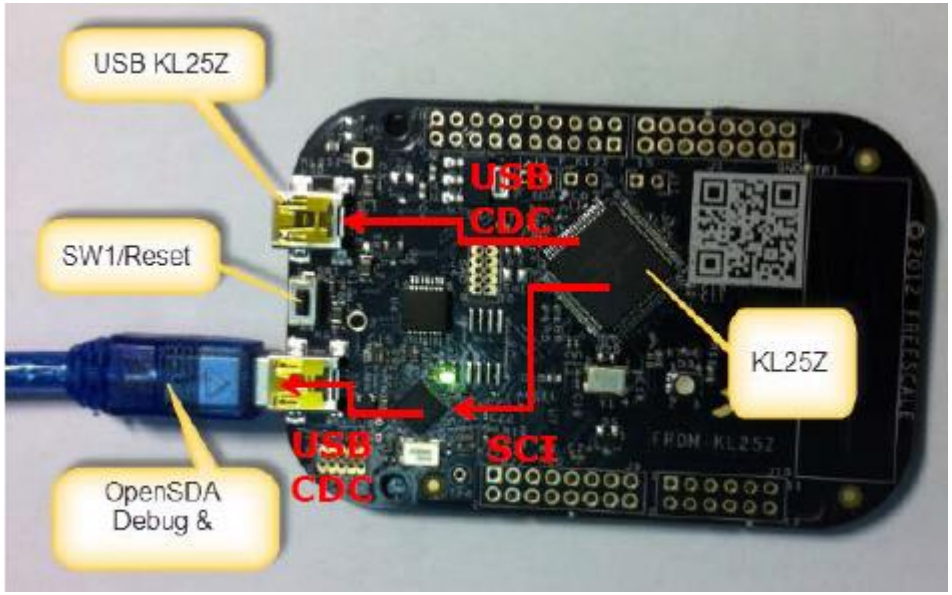
- CDC (Communication Device Class), HID, MSD, …

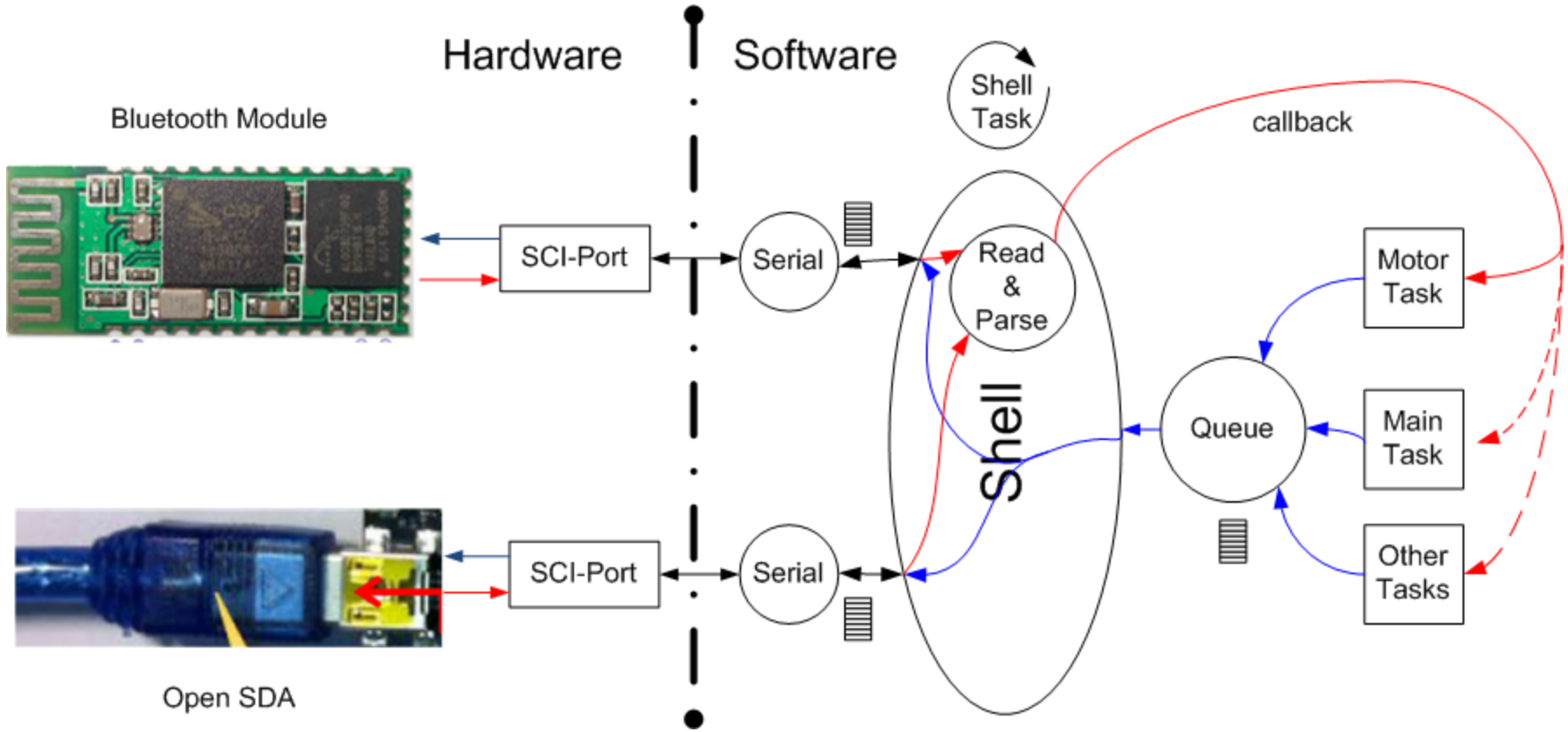
Protocol properties

- Not bound to a fixed baudrate
- Asynchron, serial, and asymmetrical (host requests to send data)

OpenSDA and USB CDC

- K20 adapts USB CDC protocol to the KL25Z SCI (UART) protocol
- Virtual COM port at the PC (USB CDC)
  - Handable like a UART protocol




Hardware
Software
Bluetooth Module
Shell Task
callback
SCI-Port
Serial
Read & Parse
Motor Task
Shell
Queue
Main Task
SCI-Port
Serial
Other Tasks
Open SDA

Shell Standard I/O

- I/O structure with callbacks

  Stdin: read char (Commands)

  Stdout: write char (Answers)

  Stderr: write char (Error)

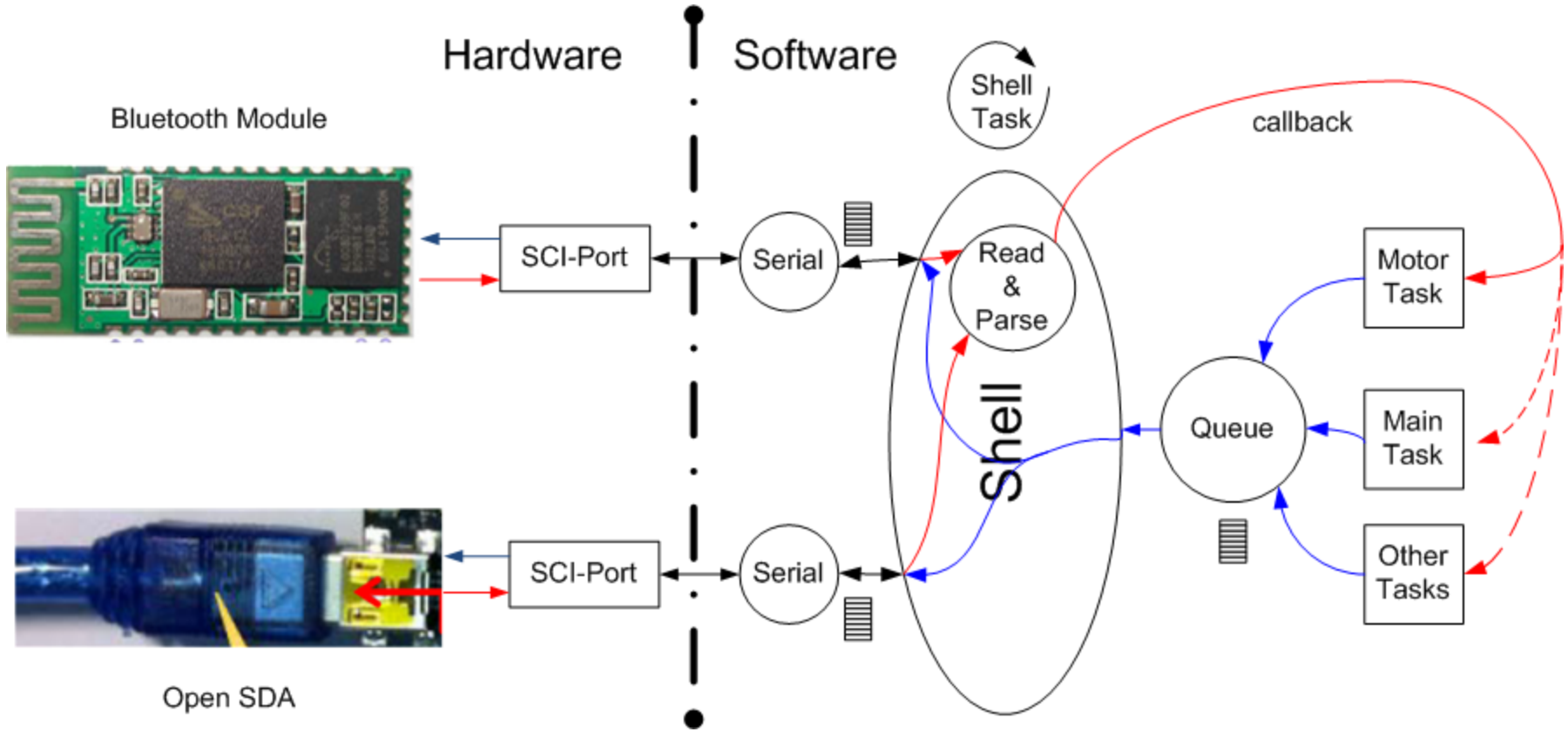
Shell Parser

```
void APP_Run(void) {
  for(;;) {
    …
    SHELL_Parse();
  }
}
```

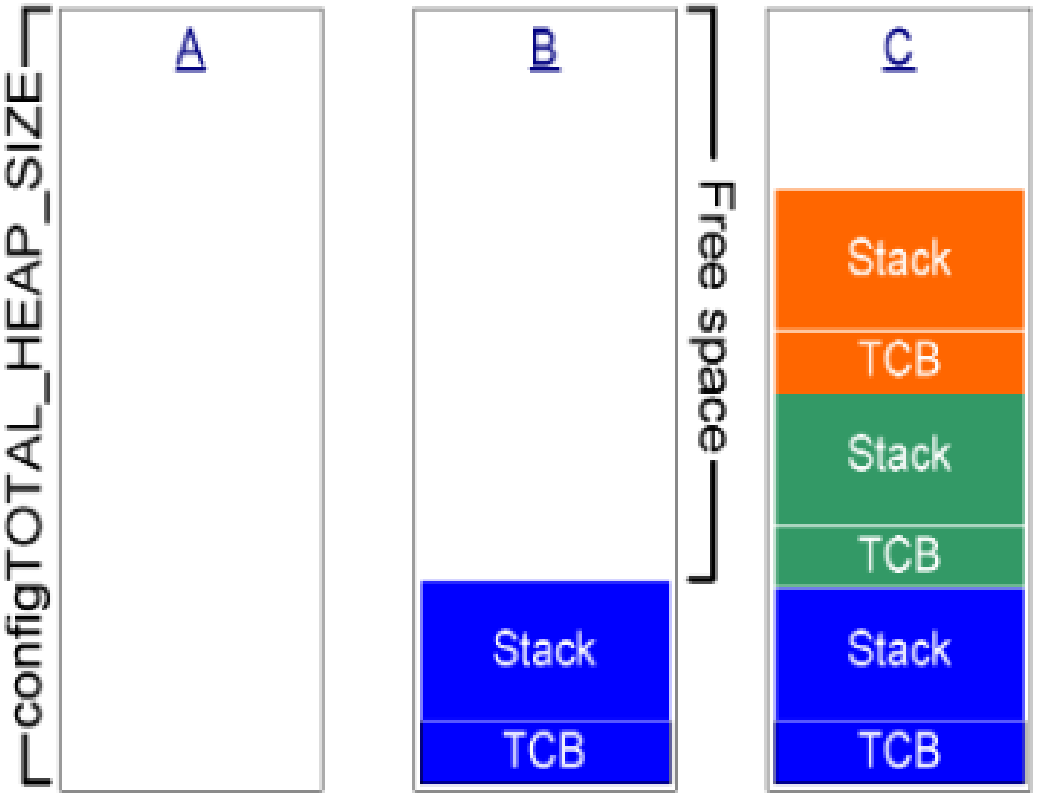
```
void SHELL_Parse(void) {
  (void)CLS1_ReadAndParseWithCommandTable(
      buf,
      sizeof(buf),
      CLS1_GetStdio(),
      CmdParserTable))
}
```

***Appends* to buffer!**
Initialize!

```
void SHELL_Init(void) {
  buf[0] = '\0'; /* init */
}
```

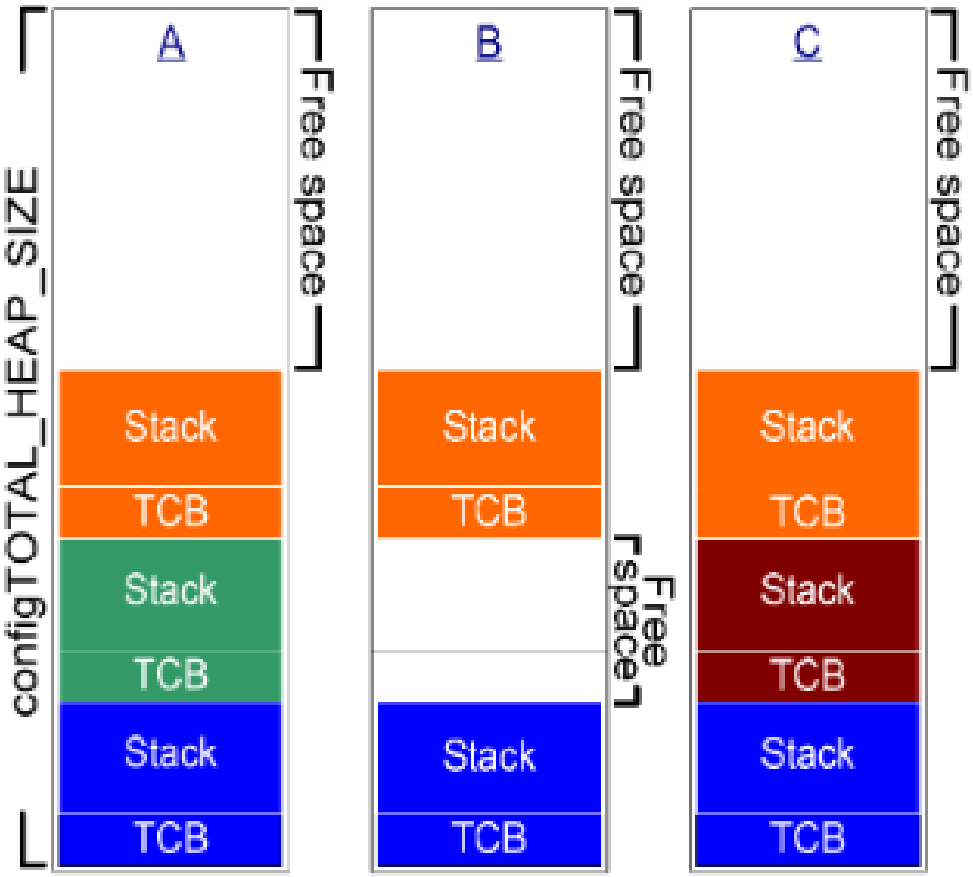

Hardware
Software
Bluetooth Module
Shell Task
callback
SCI-Port
Serial
Read & Parse
Motor Task
Shell
Queue
Main Task
SCI-Port
Serial
Other Tasks
Open SDA

Memory Scheme 1



Source: freeRTOS.org

Memory Scheme 2



Source: freeRTOS.org

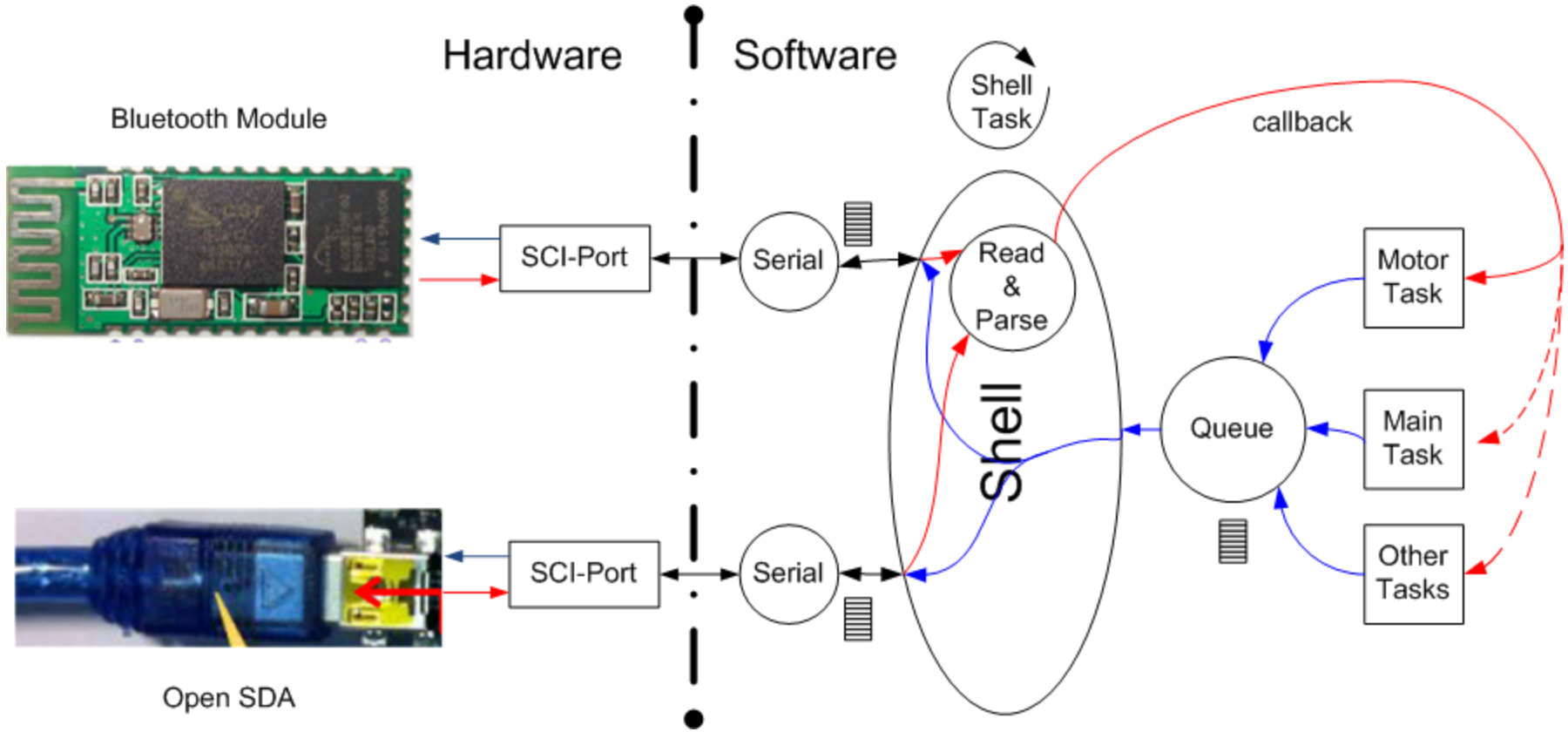
Malloc(), Free() Example

```
void *pvPortMalloc(size_t xWantedSize);
void vPortFree(void *pv);
size_t xPortGetFreeHeapSize(void);

void foo(void) {
  char_t *bufP;

  bufP = (char_t*)pvPortMalloc(sizeof("Hello"));
  if (bufP==NULL) {
    for(;;); /* ups! */
  }
  (void)strcpy(bufP, "Hello");
  /* do something with it */
  vPortFree(bufP);
}
```

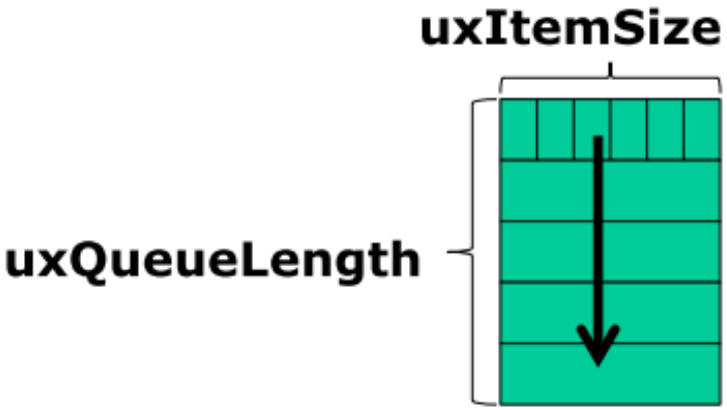
```
UTIL1_strcpy(buf, sizeof(buf), "Hello");
```


Hardware
Software
Bluetooth Module
Shell Task
callback
SCI-Port
Serial
Read & Parse
Motor Task
Shell
Queue
Main Task
SCI-Port
Serial
Other Tasks
Open SDA

Queue Create & Delete

```
xQueueHandle xQueueCreate(
  unsigned portBASE_TYPE uxQueueLength,
  unsigned portBASE_TYPE uxItemSize
);

void vQueueDelete(xQueueHandle xQueue);
```

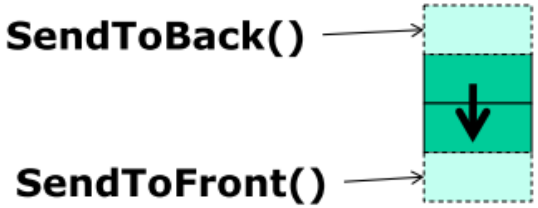


Queue Send

```
portBASE_TYPE xQueueSendToBack(
  xQueueHandle xQueue,
  const void *pvItemToQueue,
  portTickType xTicksToWait
  );
portBASE_TYPE xQueueSendToFront(
  xQueueHandle xQueue,
  const void *pvItemToQueue,
  portTickType xTicksToWait
  );
```

Hint: portMAX_DELAY

0: return immediately
>0: ticks to wait



Queue Receive & Peek

```
portBASE_TYPE xQueueReceive(
  xQueueHandle xQueue,
  void *pvBuffer,
  portTickType xTicksToWait
  );
portBASE_TYPE xQueuePeek(
  xQueueHandle xQueue,
  void *pvBuffer,
  portTickType xTicksToWait
  );
```

0: return immediately
>0: ticks to wait