



Tigers Mannheim



Embedded Programming





Aufgabe

- „Schubsen“ des Roboters über eine Wisch-Bewegung auf dem Display
- Hinzufügen eines Threads - ConsolePrint
- Ansteuern der Motoren - On/Off
- Abfangen und Interpretieren des Wisch-Events
- Hinzufügen einer Intercom Section und Commands
- Senden des Commands von Media → Main
- Erstellen eines Callbacks für die neue Intercom Section
- Weiterleiten des Events an eine Mailbox
- Abholen des Events aus dem Thread
- Implementierung des Bewegungsprofils



Coding Style

- `kleine_dateinamen.h` ↔ `kleine_dateinamen.c`
- Globale Funktionen in `CamelCase()`
- Globale Funktionsnamen entsprechen Dateinamen:
`KleineDateinamenInit()`
- Lokale Funkt. mit kleinem ersten Buchstabe: `camelCase()`
- Lokale Funktionen IMMER `static`
- Strukturen verwenden `CamelCase`
- Variablen verwenden `camelCase`
- Pointer haben ein kleines ,p': `void* pIAmAPointer`
- `#define IMMER_ALLES_GROSS 42`
- Verwendet Typen fester Größe: z.B. `int32_t`, NICHT `int`
- Globale Daten in Struktur entsprechend des Dateinamens



Coding Style

- ENGLISH please!
- `MyFunction()`
 {
 }
- Tab Einzug
- Klammern dürfen bei Einzeilern weg gelassen werden
 `if(42)`
 `printQuestion()`



CLI

- `cli.c`
- `processCommand()`
- `if(ConsoleCmpCmd("explode")) ...`
- `if(ConsoleScanCmd("params %hd" &i16) == 1) ...`
- Rückgabewert von `ScanCmd` entspricht Anzahl gefundener Parameter
- ACHTUNG: beachtet die format modifier! Besonders ,h‘
- ACHTUNG: `ConsolePrint` erfordert einen großen Thread Stack!



Hinzufügen eines Threads

- Neue datei swipe_ctrl.h/.c
- `#include "ch.h"`
- Thread Prototype:
`msg_t SwipeCtrlThread(void* params)`
- `chThdSleep (MS2ST (1000)) ;`
- `chRegSetThreadName ("SwipeCtrl") ;`
- main.c
- WORKING_AREA
 - Empfohlene Größe 4096 Byte
- `chThdCreateStatic`



Motoransteuerung

- `ctrl_vel.h`
- `void CtrlVelSetTarget(float x, float y, float w);`
- Einheit m/s, rad/s
- Lasst den Bot mit dem Thread 1s langsam vorwärts fahren, dann 1s genau so schnell rückwärts
- Startet dieses Verhalten 10s nach Einschalten des Bots und hört nach 5 mal fahren auf



Abfangen des Wisch-Events

- Media → presenter.c
`swipeCallback(TouchSwipeEvent* pSwipe)`
- Modifiziert die Funktion so dass:
 - Das Tab Wechseln per Swipe nur im unteren Bereich funktioniert
(0, 265) (240, 320)
 - Ein Swipe im Bereich (0, 30) (240, 265) auf der Konsole
ausgegeben wird (relevante Infos)



Intercom Section und Command

- lib/common/src/intercom_constants.h
- Fügt die INTERCOM_SECTION_MOVE_CTRL hinzu
- Fügt die IC_MOVE_CTRL_SWIPE Message ID hinzu
- Überlegt euch eine Struktur die alle wichtigen Daten enthält die von Media nach Main gesendet werden sollen
Name: MoveCtrlSwipe



Intercom Senden

```
IntercomHeader iHeader;  
iHeader.address = INTERCOM_ADDRESS_MAIN;  
iHeader.section = INTERCOM_SECTION_MOVE_CTRL;  
iHeader.command = IC_MOVE_CTRL_SWIPE;  
  
MoveCtrlSwipe swipe;  
// fill swipe with data  
IntercomSend(&intercom1, &iHeader, sizeof(swipe), (uint8_t*)&swipe);
```



Intercom Callback

- Main → swipe_ctrl.c

- Callback Prototyp:

```
static void moveCtrlCallback(const IntercomHeader* pHeader,  
    uint16_t dataLength, uint8_t* pData, void* pUserData)
```

- Registrierung der Callback/Section

```
IntercomClient client;  
client.pUserData = 0;  
client.rxCallback = &moveCtrlCallback;  
client.section = INTERCOM_SECTION_MOVE_CTRL;  
  
IntercomAddClient(&intercom1, client);
```



Event speichern

- Checkt pHeader und konvertiert pData zu MoveCtrlSwipe*
- Die Daten sind nur in der Intercom Callback gültig
- Kopiert daher die Daten in eine globale Variable
- Beachtet den Coding Style für globale Daten!





Mailbox anlegen

- Referenz: ChibiOS Manual im doc Ordner
- Modules → Kernel → Synchronization → Mailboxes

- Schnipsel für die globale Daten:

```
#define MOVE_CTRL_EVENT_QUEUE_SIZE 4  
Mailbox eventQueue;  
msg_t eventQueueData[MOVE_CTRL_EVENT_QUEUE_SIZE];
```

- Init:

```
chMBInit(&moveCtrl.eventQueue, moveCtrl.eventQueueData,  
        MOVE_CTRL_EVENT_QUEUE_SIZE);
```

- d



Mailbox verwenden

- chMBPost
- chMBFetch
- Siehe ChibiOS doku!
- msg_t ist nur 4 Byte groß!
- Sendet bei Ankunft der Intercom Nachricht eine Nachricht an die Mailbox
- Wartet in dem MoveCtrlTask auf eine Nachricht und gibt sie auf der Konsole aus



Fast Fertig

- Wenn in dem Task eine Nachricht ankommt, schaut in die globalen Daten nach dem letzten MoveCtrlSwipe
- Verwendet in dem Task sleep um ein sanftes Bewegungsprofil mit „ausrollen“ zu generieren
- Die Stärke sollte von den Eigenschaften des Swipes abhängen (Länge/Dauer)



Fertig!

Herzlichen Glückwunsch!

Workshop abgeschlossen!

- Wem jetzt langweilig ist, der armed den Kicker während der Fahrt und versucht den Bot auf ein Ball zu schubsen!