

Test und Spezifikationen

Inhaltsverzeichnis

- 1. Testen in der Code::Blocks Umgebung
 - 1.1. Windows Test-Spezifikation
 - 1.2. Windows Tests
- 2. Testen der ORB-Firmware
 - 2.1. Firmware Test Spezifikation
 - 2.2. Firmware Tests
 - 2.2.1. Program Upload
 - 2.2.2. Modules
 - 2.2.2.1. Devices - Servo Module
 - 2.2.2.2. Devices - Motor Module
 - 2.2.2.3. Devices - Sensor Module
 - 2.2.2.4. Memory
 - 2.2.2.5. Time
 - 2.2.3. Builtin Functions
 - 2.2.4. Error-Tests

1. Testen in der Code::Blocks Umgebung

Im Ordner "`<PATH_TO_ORB>ORB-Python\Program`" befindet sich eine Datei 'program.py'. Diese kann verwendet werden, um Funktionalitäten der Micropython-VM zu testen. Diese Datei wird von der ORB-Windows-Application geladen und ausgeführt. Hier kann man, wenn man von der Code::Blocks-Umgebung aus testen möchte, sein Programm umschreiben.

Der Python-API-Dokumentation (sphinx-docs) ist zu entnehmen, welche Funktionen bereit gestellt werden. Die erwarteten Ausgaben der Funktionen sind den Mockup-Dateien zu entnehmen. Grundsätzlich reicht hier einfaches Testen. Für diesen Zweck wurde ein Test-Ordner erstellt: `<Path_To_ORB>/ORB-Python-Exampes/Tests/Windows`.

Die Datei `_execute_(drag_test_here).bat` ist ein kleines Hilfs-Script um Tests einfach ausführen zu können. Hier ist wichtig, dass das Code::Blocks-Projekt einmal gebaut werden muss.

1.1. Windows Test-Spezifikation

- Es sollen alle Funktionen der Python-API getestet werden. Es sollen alle Funktionen einmal ausgeführt werden. Der Test-Vorgang ist wie folgt:
 - Jedes Modul bekommt einen eigenen Test
 - Funktionen mit Rückgabe-Wert: der Befehl `print(<funktions aufruf>)` ist zu verwenden
 - Funktionen mit Liste als Rückgabe-Wert: Hier wird zusätzlich auf die Listen-Elemente zugegriffen
 - Es soll die korrekte Verwendung von Objekt-Referenzen, wie z.b. bei Motoren, überprüft werden
 - Es sollen die Funktionen der Python-VM getestet werden
 - Exception-Handling und Exception-Exit-Status
 - Interrupt-Programm und Interrupt-Exit-Status
 - Der Normal-Exit-Status ist implizit in den anderen Tests vorhanden

1.2. Windows Tests

test.compilationError	Dieser Test ist für das Compile-Script. Er sollte einen Value-Error werfen. Zusätzlich kann man versuchen dieses Script als das 'program.py' in Code::Blocks zu verwenden. Kompiliert man die ORB-Firmware sollte der Build-Prozess mit entsprechendem Error abgebrochen werden.
-----------------------	--

test.exception	Dieser Fehler überprüft ob MicroPython-Exceptions als solche erkannt werden (Exit-Code 1). Zusätzlich sollte die Art der Exception ausgegeben werden. In diesem Fall 'Exception'.
test.memFault	Dieses Test-Script überprüft Memory-Allocation-Errors d.h. ungenügender Speicher im Micropython-Heap. Die VM soll mit dem Exit-Code Exception(1) beendet werden. Der Exit-Typ sollte einen Memory-Error anzeigen.
test.memory	Ein einfacher Test der Memory Funktionen. Es sollen alle Funktionen des Memory-Moduls verwendet werden.
test.monitor	Ein einfacher Test der Monitor Funktionen. Es sollen alle Funktionen des Monitor-Moduls verwendet werden.
test.motor	Ein einfacher Test der Motor Funktionen. Es sollen alle Funktionen des Motor-Moduls verwendet werden.
test.sensor	Ein einfacher Test der Sensor Funktionen. Es sollen alle Funktionen des Sensor-Moduls verwendet werden.
test.servo	Ein einfacher Test der Servo Funktionen. Es sollen alle Funktionen des Servo-Moduls verwendet werden.
test.time	Ein einfacher Test der Time Funktionen. Es sollen alle Funktionen des Time-Moduls verwendet werden.
test.userInt	Testen des User-Interrupts. Der Exit-Status sollte "2" sein und aus dem Exit-Typen sollte hervor gehen, dass ein User-Interrupt verwendet wurde.

Abbildung 11: Windows Tests

Alle Windows-Tests wurden erfolgreich durchgeführt. Das durch die Tests vorgegebene Verhalten konnte validiert werden. Die Windows-Umgebung erfüllt alle Anforderungen, die an sie gestellt wurden.

2. Testen der ORB-Firmware

Alle Firmware Tests wurden mit einem ORB, der HW-Version 0.22 durchgeführt. Es wurde ein Test-Programm für die ORB-Firmware erstellt. Dieses befindet sich unter 'ORB-Python-Examples/Tests/Firmware/firmware_test.py'. Das Test-Programm kann mit Hilfe des ORB-Monitors übertragen und ausgeführt werden. Dazu wird der Befehl `orb-util -f firmware_test.py` verwendet. Das Test-Programm bietet eine interaktive Oberfläche um alle Funktionen des Python-Interpreters zu testen. Das Test-Programm unterteilt durch eine Menü-Führung die einzelnen zu testenden Abschnitte.

2.1. Firmware Test Spezifikation

- Es soll die Programm-Übertragungen getestet werden. Dafür soll sowohl ein beliebiges C++, als auch ein Python Programm übertragen werden.
- Es sollen alle Python-Funktionen getestet werden. Der Fokus ist hier auf korrekter Parameter-Übergabe. Die Rückgabe-Werte von Funktionen werden auf dem Monitor ausgegeben. Hier sollen im Wesentlichen verschiedene Devices in den verschiedenen möglichen Konfigurationen getestet werden.
- Es sollen die Exit-Status-Ausgaben und das korrekte Handhaben von Exceptions überprüft werden. Dazu gehört das Ausgeben von Fehlermeldungen und die Wieder-Ausführbarkeit der MP-VM nach einer Exception. Es soll auch der Hard-Fault-Handler getestet werden.
- Es sollen die zusätzlich eingebundenen Buildin-Funktionen überprüft werden. Für eingebundene Module reicht hier die Verfügbarkeit, solange diese nicht im Rahmen dieses Projektes entstanden sind, so z.B. das Math-Modul.

2.2. Firmware Tests

Die Firmware-Tests sind als interaktive Tests gedacht. Es können alle Funktionen des ORB aufgerufen und ausgegeben werden. Schon durch die Möglichkeit das Test-Programm zu erstellen, ist die Funktions-Fähigkeit der ORB-Firmware zu einem großen Teil validiert. Das restliche Testen, Neben der Erreichbarkeit der Funktionen, ist das korrekte Übergeben von Parametern. Um dies testen zu können, soll das Firmware-Test-Programm verwendet werden. Es bietet die Möglichkeit an, die Python-Funktionen zu mit verschiedenen Parametern aufzurufen. Hier wurden alle Parameter-Variationen für die möglichen Funktions-Aufrufe getestet. Im Folgenden sind zusätzliche Informationen für den Vorgang der Tests. Alle im folgenden aufgezählten Tests konnten erfolgreich durchgeführt werden.

2.2.1. Program Upload

Für den Program-Upload-Test wurde das C++-Demo-Programm übertragen. Die Ausführung von diesem ist identisch zu dem C++-Programm mit alter Firmware. Die C++-Programmübertragung konnte validiert werden. Das Aufspielen des Python-Firmware-Test-Programms war auch erfolgreich. Es konnte also sichergestellt werden, dass die ORB-Firmware sowohl Python-Programme als auch C++-Programme erfolgreich übertragen kann. Außerdem ist durch diesen Test überprüft worden, dass die ORB-Firmware zwischen Python-Programm und C++-Programm unterscheiden kann. Das Python-Programm und das C++-Programm lassen sich sowohl über die ORB-Monitor-Tasten, sowie die ORB-On-Board-Tasten starten und stoppen. Die Status-LEDs des ORB werden dabei korrekt gesetzt.

2.2.2. Modules

2.2.2.1. Devices - Servo Module

Der Servo-Module-Test erlaubt es einem Nutzer alle Parameter eines Servo-Objektes einzustellen. Es sind der Port, die Geschwindigkeit und der Winkel einstellbar. Hier kann auch optional mit einer LED getestet werden, da ein Servo-Motor mit Hilfe eines PWM-Signales angesteuert wird. Es konnte validiert werden, dass Servo-Motoren konfigurierbar sind.

2.2.2.2. Devices - Motor Module

Der Motor-Test ist in drei Teile unterteilt: 'config', 'set', 'get'. Dies sind die drei Funktionen, welche durch ein Motor-Objekt verwendbar werden. Der erstellte Motor-Test erlaubt es einem Nutzer einen Lego-Motor vollständig zu konfigurieren. Einstellungen wie 'kp', 'ki', etc., also motor-spezifische Konfigurationen sind hier für den LEGO-Motor eingestellt. Diese wurden in dem Line-Follower mit anderer Konfiguration verwendet. Somit ist die Konfiguration für einen Lego-Motor und einen MakeBlock-Motor vorgenommen und überprüft worden. Alle Funktionen ließen sich validieren.

2.2.2.3. Devices - Sensor Module

Der Sensor-Test ist in drei Teile aufgeteilt: 'config', 'get', 'pre-configured'. 'get' und 'config' werden verwendet um Sensoren zu benutzen. Hier werden die Sensor-Objekt-Funktionen abgebildet. Über 'pre-configured' wird ein voreingestellter Sensor verfügbar. Hier wurde ein NXT-Light-Sensor vor-konfiguriert und durchläuft Test-Funktions-Aufrufe und Test-Konfigurationen. In diesem Test sollen verschiedene Sensoren an dem ORB angeschlossen und konfiguriert werden. Es wurden Tests für MB-Color-Sensor, MB-Ultra-Sonic-Sensor, NXT-Light-Sensor und NXT-Touch-Sensor durchgeführt. Im Wesentlichen wurde hier alle verschiedenen Funktions-Parametern für Sensoren verwendet. Dadurch wurde validiert dass diese korrekt konfigurierbar sind.

2.2.2.4. Memory

Der alternative Programmstart mit "Write Mem" kann verwendet werden um einen Wert in den Nutzer-Speicher zu schreiben. Mit dem Programm-Start "Tests" kann unter 1.Module->2.Memory die GetMemory-Funktion und ClearMemory-Funktion getestet werden. GetMemory gibt dabei den durch "Write Mem" in den Nutzer-Flash-Speicher geschriebenen Wert aus. Clear-Memory setzt den vollständigen Nutzer-Speicher zu 'xFF' zurück, darin inbegriffen den vorher geschriebenen Wert. Die Funktionalität der Memory-Funktionen konnte durch diesen Test sicher gesetzt werden.

2.2.2.5. Time

Test-Name	Test-Vorgang	Verhalten
Get Time	Ausführen.	Es wird der Wert von getTime wiedergegeben.
wait	Ausführen.	Der Nutzer wird darüber informiert, dass 2 Sekunden gewartet wird. Danach wird "Finished" ausgegeben.

Abbildung 12: Time

2.2.3. Builtin Funktions

Test-Name	Test-Vorgang	Verhalten	Notiz
Math	-	Die erste Zeile dieses Tests soll ausgeben, ob das Math-Module verfügbar ist.	
Exit	Ausführen.	Die MP-VM stoppt ihr Ausführung. Auf dem ORB-Monitor wird ein System-Exit angezeigt.	
Min/Max	Ausführen. Mit Buttons Werte auswählen.	Auf dem Monitor wird das Minimum und Maximum der beiden Werte angezeigt.	
getArg	Ausführen.	Der Arg-Wert wird ausgegeben. Dieser ist 0.	

Abbildung 13: Builtin Funktions

2.2.4. Error-Tests

Test-Name	Test-Vorgang	Verhalten	Notiz
Hard-Fault	Starte einen Motor mit dem Motor-Test. Geh zurück zum "Main Menu". Danach Hard-Fault-Test ausführen.	ORB-Firmware unterbricht Ausführung. Alle 3 Status LEDs gehen an. ORB-Firmware muss resettet werden. Der zuvor gestartete Motor wird gestoppt.	Der Hard-Fault ist schwer zu produzieren, nach Änderungen an der Firmware könnte dieser Test nicht mehr funktionieren. Die aktuelle Firmware-Version produziert diesen jedoch korrekt. Es wurde mit der Hardware-Version 0.22 getestet.
Memory-Error	Ausführen	Die MP-VM unterbricht die Ausführung. Es wird eine Exception-Message auf dem ORB-Monitor ausgegeben. Diese weist auf einen Memory-Error hin.	
User-Exception	Ausführen	Die MP-VM unterbricht die Ausführung. Es wird eine Exception-Message auf dem ORB-Monitor ausgegeben. Diese weist	

Test-Name	Test-Vorgang	Verhalten	Notiz
		auf eine User-Exception hin.	
User-Interrupt	Ausführen, danach muss der Stop-Button des ORB gedrückt werden.	Die MP-VM unterbricht die Ausführung. Es wird der 'User Interrupt' auf dem ORB-Monitor ausgegeben.	

Abbildung 14: Error-Tests