



Xcom-API

version **v1.0** python **3.4, 3.5, 3.6** license **GNU GPL v3** documentation **wiki** download **zip**

i Introduction

This project is about an API (**A**pplication **P**rogramming **I**nterface) designed as a python-class, which is used to encode and decode byte-frames for communication between a Xtender-System and a PC, especially a Raspberry Pi with a Raspicomm extension module. The class also contains methods for error and frame-flag decoding and methods to get object or class informations.

Note: The API and the Test-Program was created with Microsoft's integrated development environment (IDE) Visual Studio Code (VSC). This is an Open-Source IDE from Microsoft for many different programming-languages for Windows, Linux and Mac with an easy to use git integration. So you can easily add this project to VSC by cloning this Git-Repository. Of course you can open the python-files with other IDE's you prefer too.

▲ Git-Repository-Link: http://es-lab.de:60085/mueller_to/Xcom-API

Project-Folders

Xcom-API

This class is used to encode byte-frames, which are needed to communicate with Xtender-Moduls over the RS232-BUS with the Xcom-232i. After encoding the byte-frame, you can use the module "pyserial" to communicate with the Serial-BUS and to put the byte-frame into the write-method of "pyserial". You can also decode read-frames of the Serial-BUS with this class.

The class contains methods to generate read- or write-frames, and methods to decode frames. There are pre-defined types, properties and formats for the most important parameter- and information-numbers. This reduces the effort for generating byte-frames, but there are extended methods to encode byte-frames with other parameter- and information-numbers too, that you can find in the documentation of the Xcom-232i. The class also contains methods for error and frame-flag decoding and methods to get object or class informations.

More informations about the class and their methods you can find on the Wiki-Page.

Xcom Test Program

This program demonstrate the functionality of the Xcom_API-class. To use this program a Raspberry Pi with a Raspicomm extender module and a RS232-bridge called Xcom-232i is required.

More information about the hardware-setup and the usage of the program you can find on the Wiki-Page.

Xcom Test-Loop Program

This program demonstrate the functionality of the Xcom_API-class. It sends permanently a request for a known object_id and display the answer from the Xtender-System. To use this program a Raspberry Pi with a Raspicomm extender module and a RS232-bridge called Xcom-232i is required.

More information about the hardware-setup and the usage of the program you can find on the Wiki-Page.

Raspberry Pi Image

Here you can find an image of the Raspberry Pi SD-Card, which i have used for testing. The COM-Port and GPIO setup in this image is already finished and ready to use.

On the Wiki Page you can find informations about: "How to copy the image to a SD-Card" and about "How to setup the COM-Port and GPIOs" on your own Raspberry-Distribution.

Hardware-Configuration

Hardware Information

The following table shows the needed hardware components and their usage and give you links to the product information and documentation. These components are necessary if you want to use the Xcom-Test-Program and Xcom-Test-Loop-Program.

Component	Function	Product Information	Documentation
Raspberry Pi 3	single board computer, on which is running the program	🔗	🔗
RasPiComm	expansion board for the Raspberry Pi, which provides a RS232-Port over the GPIOs (UART) of the Raspberry Pi	Product not available anymore	Product not available anymore
Xcom-232i	CAN to RS232 Bridge, which is used to communicate with the Xtender-Modules	🔗	🔗 🔗
RCC-02	remote control and programming unit for displaying the status informations and to program parameters of the Xtender-Modules	🔗	🔗
Xtender-Series	is used as Battery charger and as DC to AC converter and to supply electrical power	🔗	🔗

Hardware Setup

The picture below shows the connection of the Hardware components.



ToDo-List

The following checklist shows the actual state of this project.

- create Git-Repository
- upload project-files
- finish "Xcom_API"
- finish "Xcom_Test_Program"
- finish "Xcom_Test_Loop_Program"
- finish documentations/Wiki-Page

**Hochschule Anhalt | Anhalt University of Applied Sciences | Special Field 6 EMW
Xcom-API**

Tobias Müller, M. Eng.

✉ Tobias.Mueller@HS-Anhalt.de

© es-lab.de, 16.05.2018