acontis technologies GmbH

SOFTWARE

EC-Simulator

Rust Programming Interface

Version 3.2

Edition: November 6, 2025

Contents

1 Introduction

The Rust Wrapper provides a Rust interface to use EC-Master, EC-Simulator and RAS Client/Server.

1.1 Requirements

Rust v1.90.0 and above

- Download “rustup” and install Rust

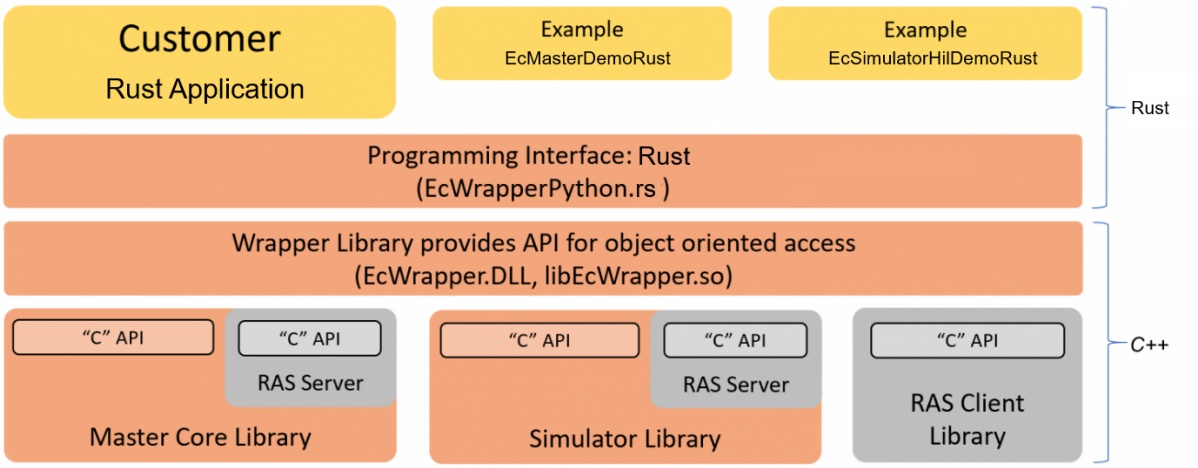
Windows (x86/x64)

- Microsoft Windows 10 and above
- Microsoft Visual C++ 2015 Runtime

Linux (x86/x64/ARM)

- Ubuntu 18.04 and above

1.2 Architecture



The architecture contains 4 basic layers:

Customer Rust application or our examples (EcMasterDemoRust, …)

- Demo application, written in Rust

Programming Interface (EcWrapperRust)

- Provides an object oriented API written in Rust

Wrapper Library (EcWrapper)

- Native wrapper library, which provides API for object oriented access

Native Libraries

- Master Core Library
- Simulator Library
- RAS Client Library

2 Programmers Guide

2.1 Sample Application

There is currently 1 script available:

EcMasterDemoRust.bat
Starts the console demo application

2.2 Sample Code

The Rust demo application contains of 3 modules:

EcDemoApp.rs:
Console demo application

EcLogging.rs:
Logging module, which writes asynchronous messages to console, file, …

EcUtil.rs:
Utility module, which contains some helper functions

2.3 Wrapper

2.3.1 Modules

The Rust Wrapper contains of 5 modules:

EcWrapperRust.rs

class CEcWrapperRust
EC-Wrapper base class

class CEcMasterRust
EC-Master

class CEcMasterMbxGatewayClientRust
Mailbox Gateway Client for EC-Master

class CEcMasterMbxGatewayServerRust
Mailbox Gateway Server for EC-Master

class CEcSimulatorRust
EC-Simulator

class CEcSimulatorRasServerRust
RAS Server for EC-Simulator

class CEcRasClientRust
RAS Client for EC-Master / EC-Simulator

EcMotionRust.rs

class CEcMotionRust
EC-Motion interface

`EcWrapperRustTypes.rs`
: Rust types

`EcWrapper.rs`
: C Rust interface (internal)

`EcWrapperTypes.rs`
: C Rust types (internal)

2.3.2 Return code vs. exception handling

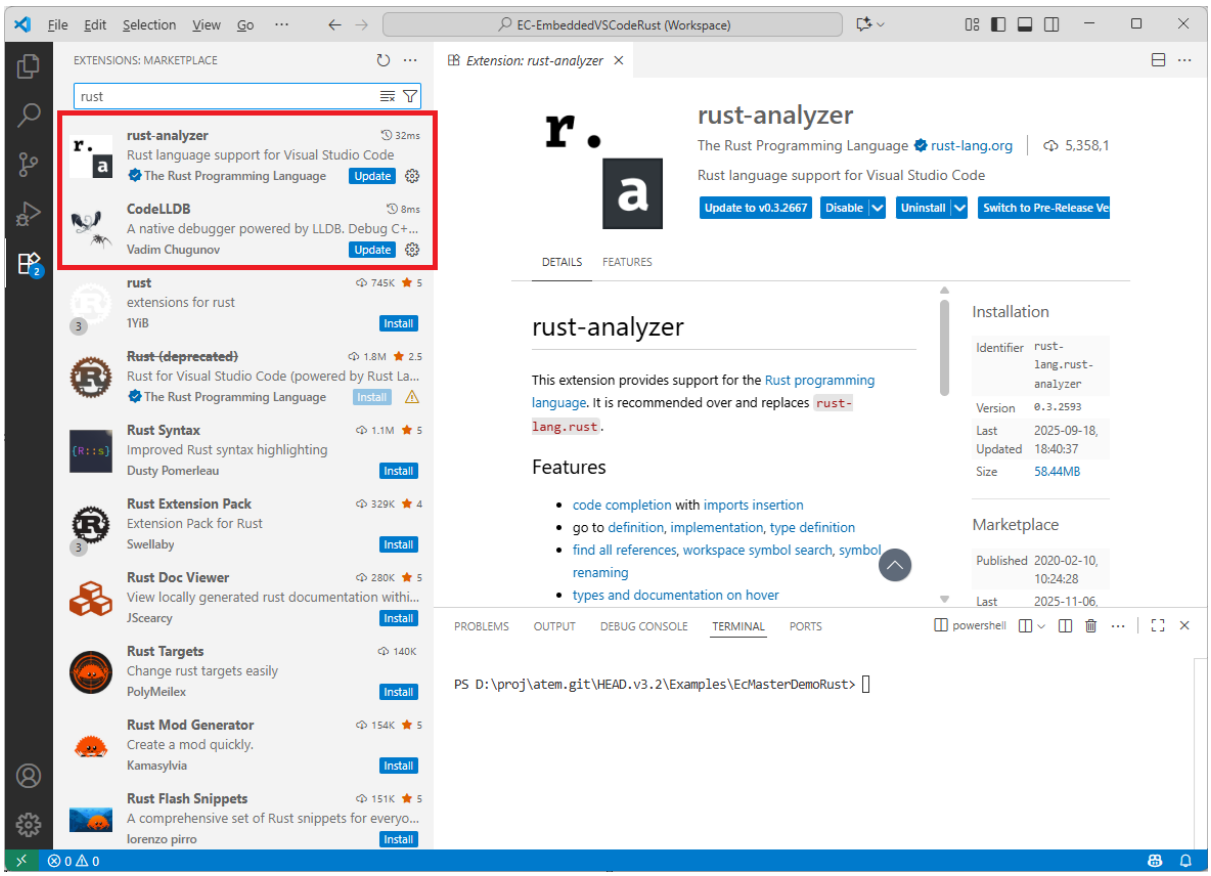
The most of all API functions returns a return code for error handling. This behaviour can be changed to throw an exception in error case by simply setting:

```
CEcWrapperRust_EnableExceptionHandling = true // default is false
```

2.4 Supported IDEs

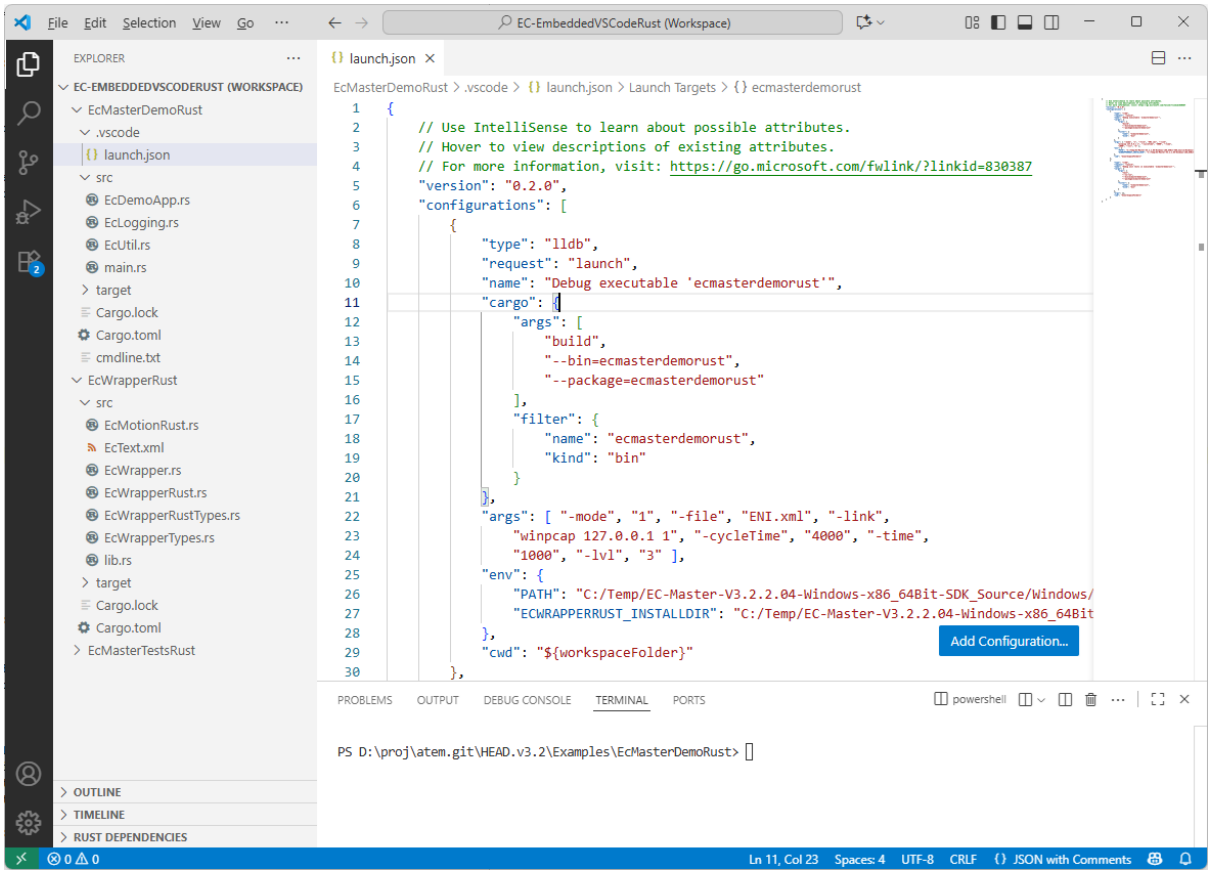
2.4.1 Visual Studio Code

Install rust extension by open extension tab and enter *rust*:

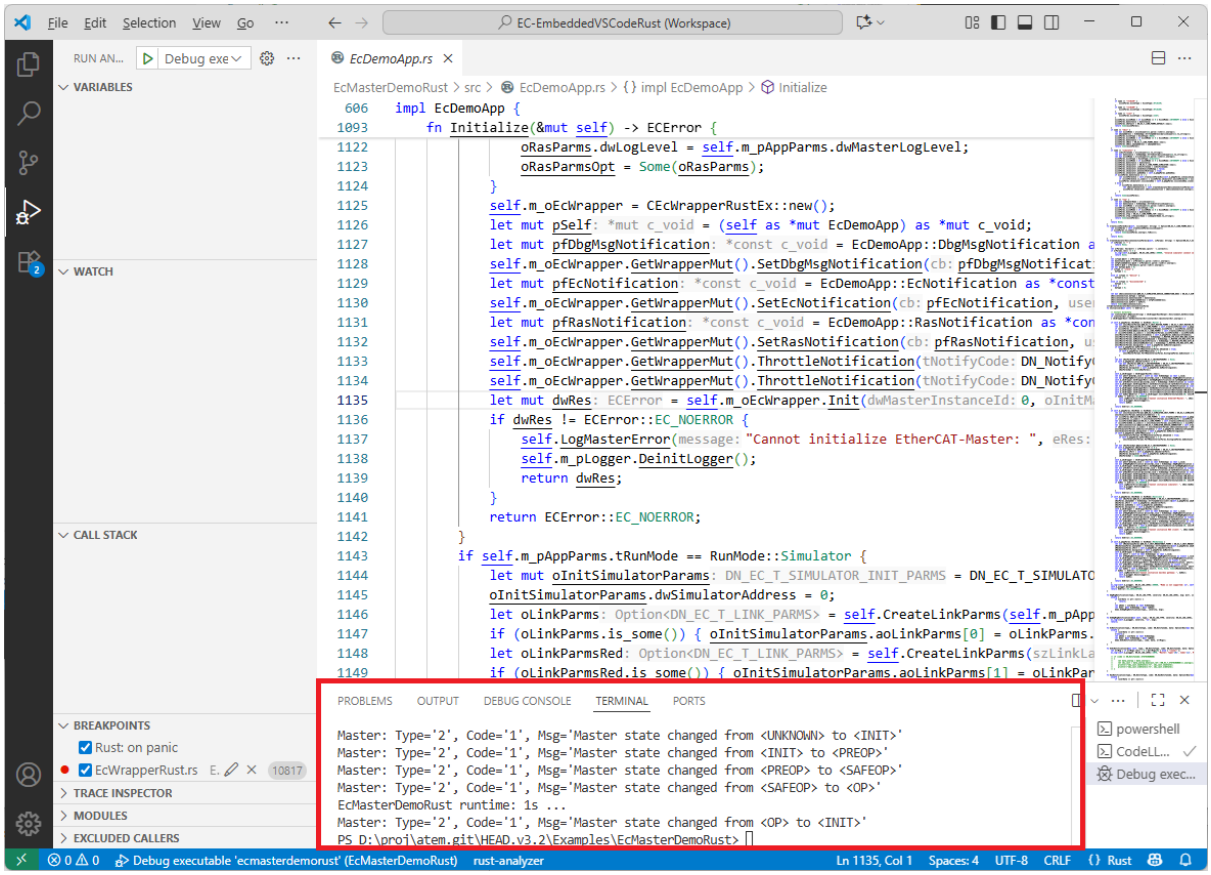


Open package folder e.g. `EC-Master-V3.2.2.04-Windows-x86_64Bit-SDK_Source` and configure the `launch.json`:

```
{
    "version": "0.2.0",
    "configurations": [
        {
            "type": "lldb",
            "request": "launch",
            "name": "Debug executable 'ecmasterdemorust'",
            "cargo": {
                "args": [
                    "build",
                    "--bin=ecmasterdemorust",
                    "--package=ecmasterdemorust"
                ],
                "filter": {
                    "name": "ecmasterdemorust",
                    "kind": "bin"
                }
            },
            "args": [ "-mode", "1", "-file", "ENI.xml", "-link", "winpcap 127.
↪0.0.1 1", "-cycleTime", "4000", "-time", "1000", "-lvl", "3" ],
            "env": {
                "PATH": "C:/Temp/EC-Master-V3.2.2.04-Windows-x86_64Bit-SDK_
↪Source/Windows/x64/",
                "ECWRAPPERRUST_INSTALLDIR": "C:/Temp/EC-Master-V3.2.2.04-
↪Windows-x86_64Bit-SDK_Source/Windows/x64/"
            },
            "cwd": "${workspaceFolder}"
        },
    ]
}
```



Start debugging and the demo output will be written into the terminal:



3 FAQ

I installed Rust and the demo crashes with strange errors. What can I do?

> This might be a problem of mixing x86 with x64 binaries. Verify that if you have installed the Rust runtime for x64 bit, please install also EC-Master for x64 bit.