

|  |  |
| --- | --- |
|  | **imotion** |
|  |  |
|  | OnBoard Computer |
|  |  |
|  | Migration 2015 |
|  |  |
|  | Implementation Document |
|  |  |
|  | Version: 0.1 |
|  | State: Draft |
|  | Classification: Internal use only |
|  | Author: WES |
|  | Creation date: 2015-07-01 |
|  | Repository: |
|  | Gorba AG  Sandackerstrasse  9245 Oberbüren  Switzerland |

**Table of contents**

[1 Introduction 4](#_Toc423510804)

**Modification management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Modifications** | **State** |
|  |  |  |  |  |  |

**Review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Release**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

## Scope

This document describes the intended steps to migrate the Gorba.Motion.Obc software projects from the “France” SVN repository to the Gorba TFS and integrate it with the basic applications (System Manager, Update and Hardware Manager) and the Medi protocol.

## Intended Audience

This document is written for imotion software developers. It requires knowledge of Medi and the basic imotion applications.

# Projects

* Autostart: replaced by SystemManager (exists)
* Bus: create new Bus.exe in C# (in progress)
* BusControl: obsolete except for bus driving event (HWM)
* Common: t.b.d.
* Eci: rewrite in C# in Gorba.Common.Protocols and use it in Bus.exe (C#) (in progress)
* Edi: Messages: Obu.Common; rest is now Medi
* Erg: check if needed
* GprsAutoUpdate: replaced by Update (exists)
* GprsNetup: check if needed (if so, goes to HWM)
* GpsPilot: rewrite in C# in HWM
* Ibis: rewrite in C# in a separate application (in progress)
* Ira: check if needed (Biel yes, Fribourg no)
* KeepUp: we could create a small console app for this (used when SM is not running), not priority
* Ltn: not used in VM.x
* MemoryControl: implement in HWM (configurable)
* Qnet: check if needed (Biel yes, Fribourg no)
* Setup: replaced by Update (exists)
* Sign: Bus.exe must generate all Ximple for Gorba exterior signs, other signs: check if needed
* StartUp: replaced by SystemManager (exists)
* Terminal: port to new architecture (in progress)
* Tft: not used outside France
* Tools: check if needed (low priority)
* TrafficLight: check if needed (Biel yes, Fribourg no)
* Vt3: old GUI, should not be required anymore

# Inputs and Outputs

General rule: there are (software) inputs for all states, outputs can be configured to replicate an input value

* Software States:
  + GpsCoverage (1 = true, 0 = false), done in HWM
  + GprsState (1 = true, 0 = false), to do in HWM
  + BlockState (1 = true, 0 = false), done in Terminal
  + TripState (1 = true, 0 = false) , done in Terminal
  + BufferState (1 = true, 0 = false) , done in Terminal
  + BusDriving (1 = true, 0 = false), to do in HWM
* Hardware I/Os (in HWM):
  + DoorsOpen (1 = true = open, 0 = false = closed)
  + StopRequested (1 = true, 0 = false)

# Terminal.Control

Move to Terminal.Control:

* Keyboard
* LedControl
* ScreenUtil

Done:

* Stop Requested: I/O instead of messages
* GPRS State: I/O instead of messages

Later:

* remove ML if possible
* Change all timers (except UI) to ITimer
* All time spans (config and others) to TimeSpan
* Move native methods to Win32
* Remove all Thread.Sleep()

# Bus.exe

## Configuration

* ECI
  + Host
  + Port
  + Protocol? (TCP/UDP)
  + ~~Log level for messages to BGS~~
* Reboot when no ack
* Service type (Girouette, Modele, Com)
* Use Calendar.ini (SERVICES > SCHED)
* Start of the day

## Tasks

* Every 500 ms:
  + Handle GPS data
* Every second:
  + Handle GPS position (GestionPosition)
  + GestionVE4?
* Every minute:
  + Send config over Medi if initialized (SendEHTrameConfig)
  + Check if keep-alive ack was received via ECI (restart socket/modem if not, reboot after 10 missing acks)
* Every 2 minutes:
  + Send a keep-alive via ECI
* Initialize once GPS time is available
* Load service information (now from persistence)
* Send GPS position every 20 seconds over ECI

## Obsolete Tasks Done by Bus.exe

* Reboot regularly (but we might have to add a random delay)

# Hardware Manager

* SetBrightness()
* BusDriving (see BusControl project, only VM.cu)
* Code:

SystemTimeUtil.SetGmtTimezone(false);

// register the APN

string apn = DeviceFactory.Instance.VehicleConfig.Apn;

if (!String.IsNullOrEmpty(apn))

{

Registry.SetValue(@"HKEY\_LOCAL\_MACHINE\ExtModems\MC75i-VDU2\Init", "3",

string.Format("AT+CGDCONT=1,\"IP\",\"{0}\"<cr>", apn));

// delete 4 ?

}

if (DeviceFactory.Instance.VersionDriver.HwType == HwType.VmC) // IO non

implémentés sur vm.cd (voir VmcdFactory.CreateInputOutput)

{

if (File.Exists(@"\flashfx disk\LAN.xml"))

{

// make IP settings if F5 is not pressed

InputOutput io = DeviceFactory.Instance.InputOutput;

if (io.GetInput(25).Read()) // WES: hack: 25 = Inputs.IN\_F5

{

Logger.Add(LogType.Info, "", "Lan.xml found");

LanSettings lanSettings = LanSettings.ReadFromFile(@"\flashfx disk\LAN.xml");

if (lanSettings != null)

{

Logger.Add(LogType.Info, "", "Rebinding to " + lanSettings.IpAddress);

lanSettings.Apply();

}

}

}

}