

# LABORATORY OF EMBEDDED SYSTEMS ARCHITECTURE

## Exercise 2

### Topic: Installing Microsoft Windows and drivers on the Intel Galileo Gen2.

#### 1. Laboratory stand

The laboratory stand consists of a PC, Intel evaluation board Galileo Gen2, a set of wiring and basic sensors cooperating with the Base Shield from the Grove Starter Kit Plus Intel IoT Edition (Fig. 1).



Fig.1. A set of sensors and wiring includes in Grove Starter Kit Plus Intel IoT Edition

Grove Starter Kit Plus for IoT Intel Edition includes also:

- transition plate (Base Shield) to enable connection sensors to the Intel Galileo Board,
- alphanumeric LCD display (2 lines of 16 characters) with backlight RGB,
- push button,
- rotary potentiometer,

- electromagnetic relay module,
- buzzer module,
- microphone module,
- light sensor,
- temperature sensor,
- touch sensor,
- servo,
- three LED diodes (blue, green and red) cooperating with LED module Socket Kit,
- battery adapter,
- information brochure,
- 8GB SD card.

To do the exercise Visual Studio 2013 Express for Windows Desktop with Windows Developer Program for IoT (<https://connect.microsoft.com/windowseembeddedIoT/Downloads>) is needed. The mentioned additive installs on PC system software (Galileo Watcher), which allows communication Visual Studio 2013 Express with Intel Galileo platform via Ethernet network.



Fig. 2. Micro SD card slot used for laboratory

## 2. Windows system installation on Intel Galileo Gen2 platform.

Microsoft Windows system installation on Intel Galileo platform should start with getting (after registration on Microsoft Connect site) the relevant software from Microsoft company site: <https://connect.microsoft.com/windowseembeddedIoT/Downloads>.

The required software, which is subject to automatic update:

1. The image of Windows system in WIM file format (Windows Developer Program For IoT – Windows Image):

***9600.16384.x86fre.winblue\_rtm\_iotbuild.150309-310\_galileo\_v2.wim***(available 2015-03-12)

- Script, that allows to prepare SD card from Windows system image (Windows Developer Program For IoT – apply-bootmedia.cmd):

**apply-BootMedia.cmd**

(available 2015-02-05)

Unfortunately Windows system image to the Galileo platform provided by Microsoft does not contain drivers for wireless network card. The drivers should be installed independently by using the **Deployment Image Servicing and Management tool (DISM)**. To do this the necessary drivers should be downloaded from the Internet, for example from:

<https://downloadcenter.intel.com/download/24575/Intel-PROSet-Wireless-Software-and-Drivers-for-IT-Administrators>.

The drivers should be dedicated for 32-bit version of Windows 8.1, that means **Wireless\_17.13.11\_De132.zip** should be downloaded.

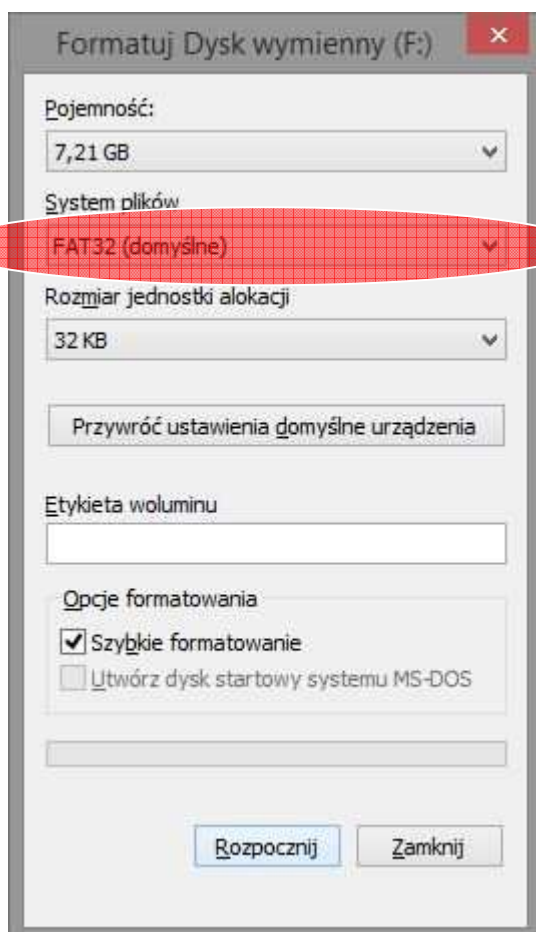


Fig.3. Formatting micro SD card

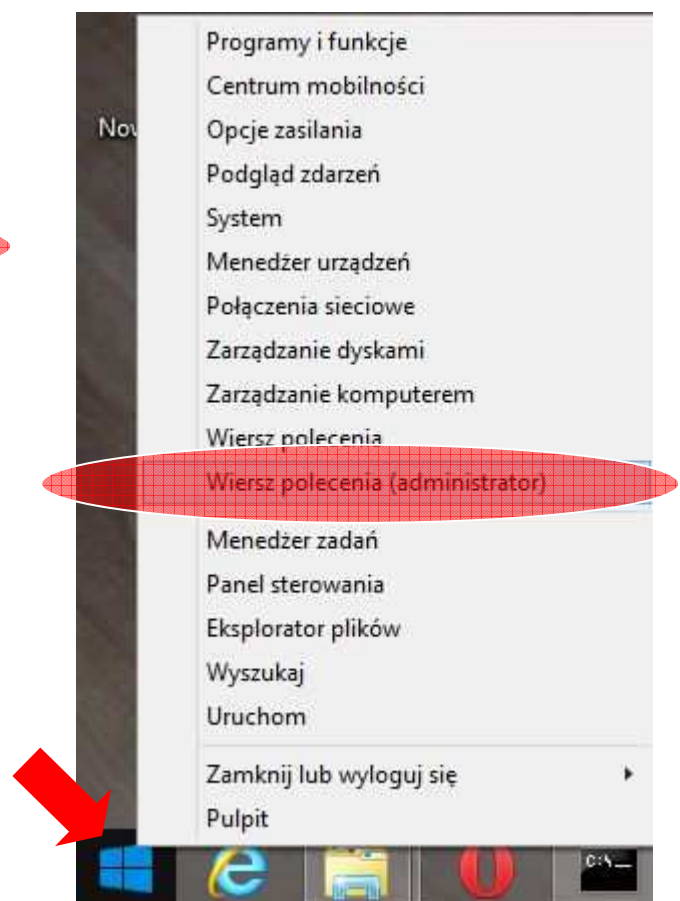


Fig.4. How to run the Command line

Downloaded files should be placed in **SW folder on disk E**. In case problems with downloading files, instructor will indicate alternative sources of required software. Downloaded files should be organized in a clear structure of folders:

**Image files -> E:\SW\Image**

**Drivers' files -> E:\SW\Drivers**

**Mounted image files -> E:\SW\Offline**

Insert carefully the micro SD memory card included in Grove Starter Kit Plus Intel IoT Edition into memory card reader (Fig. 2). Format micro SD card using FAT32 files system (Fig. 3).

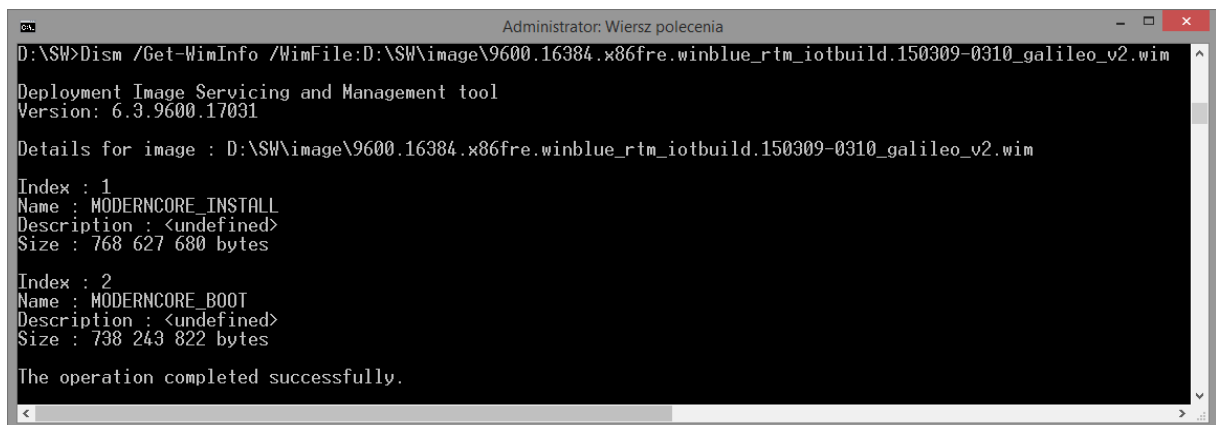
After formatting the SD card start a command line with administrator privileges. You can do this by right-clicking on the icon in the lower left corner of the desktop Windows 8.1 and then selecting: Command line (administrator) (Fig. 4). Navigate to the folder SW with the command:

```
e:  
cd SW
```

In the command line by typing Dism/Get-WimInfo you can get information about Windows image. For sample cited folder structure syntax of this command takes the form:

```
Dism /Get-WimInfo /WimFile:E:\SW\image\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim
```

The above command should result in displaying information as shown in Figure 5.



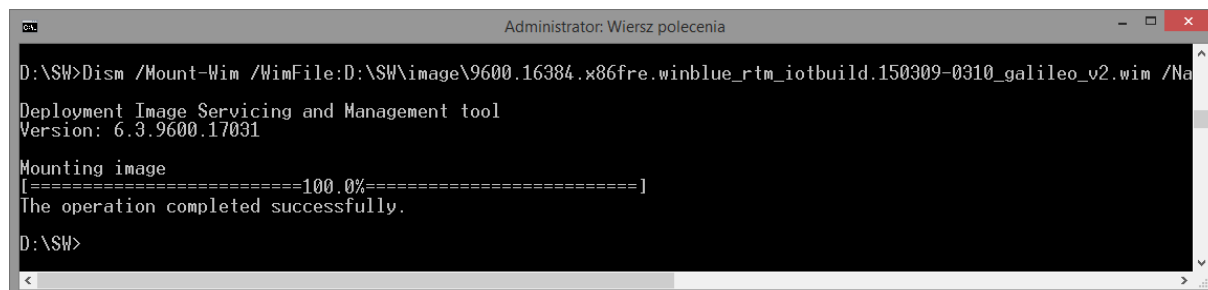
```
Administrator: Wiersz polecenia  
D:\SW>Dism /Get-WimInfo /WimFile:D:\SW\image\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim  
Deployment Image Servicing and Management tool  
Version: 6.3.9600.17031  
  
Details for image : D:\SW\image\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim  
Index : 1  
Name : MODERNCORE_INSTALL  
Description : <undefined>  
Size : 768 627 680 bytes  
  
Index : 2  
Name : MODERNCORE_BOOT  
Description : <undefined>  
Size : 738 243 822 bytes  
  
The operation completed successfully.
```

Fig.5. Window with information about owned image of Windows 8.1.

The next step is to mount the image files to the Offline folder and adding required device drivers (eg. Ethernet adapter Intel® Centrino® Wireless-N 135). This activity can be done by issuing the following command:

```
Dism /Mount-Wim /WimFile:E:\SW\image\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim /Name:"MODERNCORE_INSTALL" /MountDir:E:\SW\offline
```

The correct execution of this command will be reported by a message as in Figure 6.



```
Administrator: Wiersz polecenia
D:\SW>Dism /Mount-Wim /WimFile:D:\SW\image\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim /Name:"MODERNCORE_INSTALL" /MountDir:E:\SW\offline
Deployment Image Servicing and Management tool
Version: 6.3.9600.17031
Mounting image
[=====100.0%=====]
The operation completed successfully.
D:\SW>
```

Fig.6. Window informing about the successful mounting of Windows 8.1 image.

After mounting the Windows in folder **E:\SW\Offline**, folder structure and files appear as in any Windows system (Fig. 7).

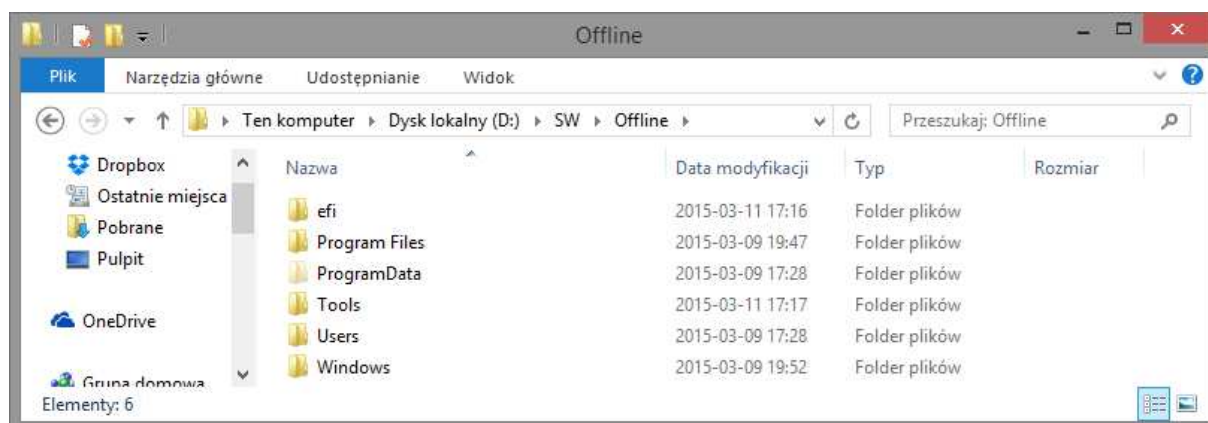


Fig.7. Folder structure of Windows 8.1 on the Intel Galileo platform.

Adding drivers to the mounted Windows image can be performed using the command **Dism/Add-Driver**. For the proposed folder structure, it will take the form of:

```
Dism /Image:E:\SW\offline /Add-Driver /Driver:E:\SW\drivers\driver_name.inf
```

where, as the name of the driver the names of the files with the extension \*.inf must be given. In case the network card drivers Intel® Centrino® Wireless-N 135, there are three files with the extension \*.inf. Therefore it is more comfortable to point out once all the drivers from **E:\SW\Drivers** folder. This can be done by enter the following command:

```
Dism /Image:E:\SW\offline /Add-Driver /Driver:E:\SW\driver\ /Recurse
```

After issuing this command **Deployment Image Servicing and Management tool (DISM)** automatically detects all the drivers located in the directory specified in the command and installs them in the mounted Windows image. The window shown in Figure 8 informs about the correctness of this action.

```

Administrator: Wiersz polecenia
D:\SW>Dism /Image:D:\SW\offline /Add-Driver /Driver:D:\SW\driver\ /Recurse

Deployment Image Servicing and Management tool
Version: 6.3.9600.17031

Image Version: 6.3.9600.16384

Searching for driver packages to install...
Found 3 driver package(s) to install.
Installing 1 of 3 - D:\SW\driver\Netwbn02.INF: The driver package was successfully installed.
Installing 2 of 3 - D:\SW\driver\Netwen00.INF: The driver package was successfully installed.
Installing 3 of 3 - D:\SW\driver\Netwsn00.INF: The driver package was successfully installed.
The operation completed successfully.

```

Fig.8. Window informing about the correct installation of the drivers in the system Windows 8.1 on Intel Galileo.

Mounted and enriched with the appropriate drivers Windows 8.1 image should be unmounted with the command:

```
Dism /Unmount-Wim /MountDir:E:\SW\offline /Commit
```

This action should be confirmed with message as in Figure 9.

```

Administrator: Wiersz polecenia
D:\SW> Dism /Unmount-Wim /MountDir:D:\SW\offline /Commit

Deployment Image Servicing and Management tool
Version: 6.3.9600.17031

Image File : D:\SW\image\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim
Image Index : 1
Saving image
[=====100.0%=====]
Unmounting image
[=====100.0%=====]
The operation completed successfully.

D:\SW>

```

Fig.9. A window informing that image of Windows 8.1 was correctly unmounted.

To place prepared earlier Windows image to an SD card, use script *apply-bootmedia.cmd* provided by Microsoft Company.

Check, under which letter in Windows SD card reader is seen. Prepare bootable SD card with Windows for Galileo platform issuing the command:

```
apply-bootmedia.cmd -destination {path to SD card} -image {path to Windows system image in WIM format } -hostname {host_name} -password {password}
```

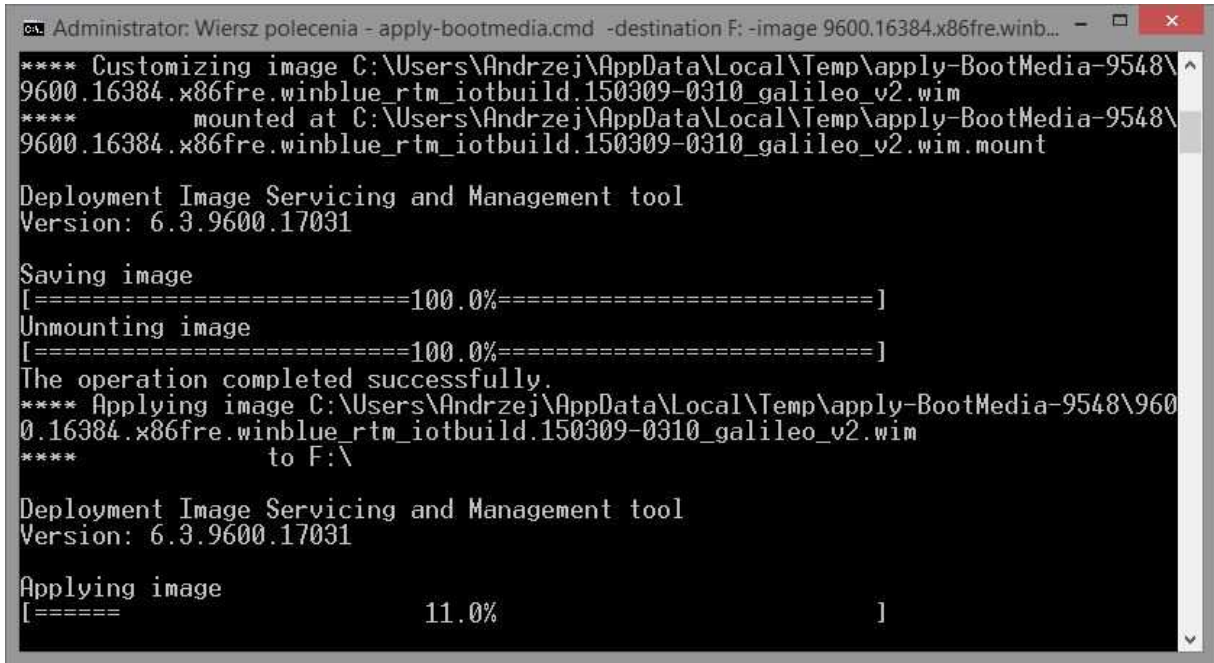
As the host name unique string of the characters in Ethernet network should be written (eg. The name or initials of performing the exercise). Host name and password should be written down in the protocol they will be needed later in a laboratory. Examples of calling the script *apply-bootmedia.cmd* might look like this:

```
apply-bootmedia.cmd -destination F: -image
D:\SW\image\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim -hostname
Kowalski -password Quark
```

**Note 1!**

The Tab key may make it easier to type the file name \*.wim.

Depending on the speed of your computer procedure for creating a bootable SD card with Windows may take several to tens of minutes. The progress in the preparations SD card will be presented in the Command Line window (Fig. 10).



```
Administrator: Wiersz polecenia - apply-bootmedia.cmd -destination F: -image 9600.16384.x86fre.winb...
**** Customizing image C:\Users\Andrzej\AppData\Local\Temp\apply-BootMedia-9548\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim
****          mounted at C:\Users\Andrzej\AppData\Local\Temp\apply-BootMedia-9548\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim.mount

Deployment Image Servicing and Management tool
Version: 6.3.9600.17031

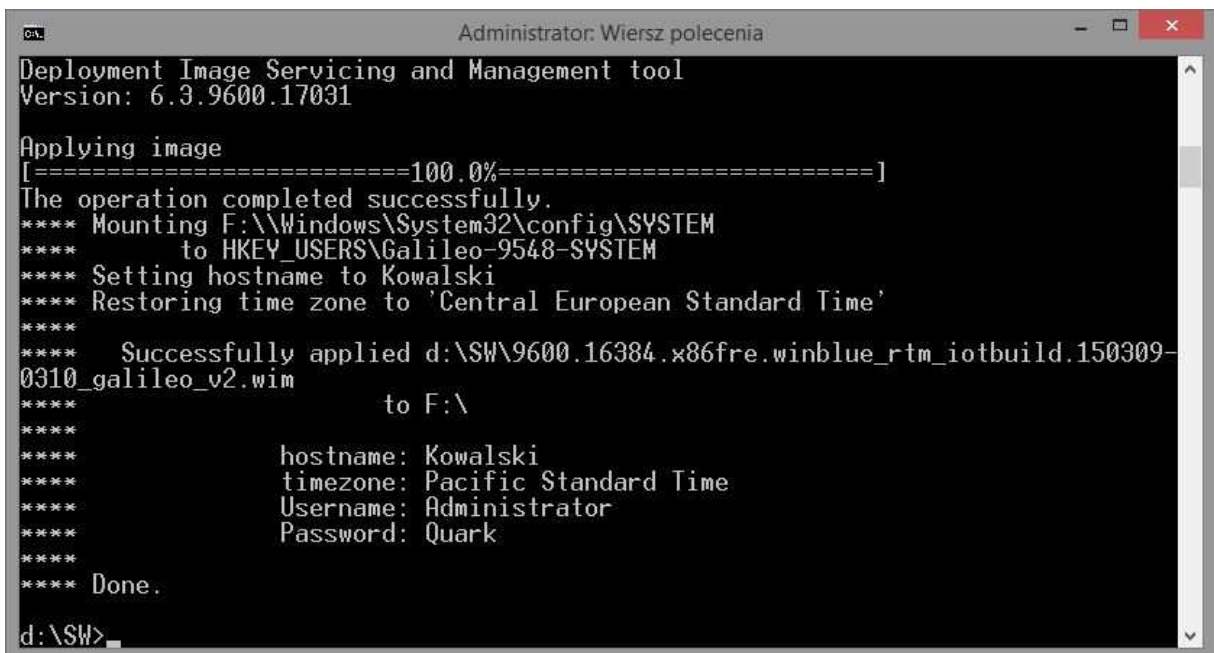
Saving image
[=====100.0%=====]
Unmounting image
[=====100.0%=====]
The operation completed successfully.
**** Applying image C:\Users\Andrzej\AppData\Local\Temp\apply-BootMedia-9548\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim
****          to F:\

Deployment Image Servicing and Management tool
Version: 6.3.9600.17031

Applying image
[=====          11.0%          ]
```

Rys.10. Progress preparation procedure SD card with Windows for Intel Galileo.

After completion of the creation of a bootable SD card with Windows you will get the message as in Figure 11.



```
Administrator: Wiersz polecenia

Deployment Image Servicing and Management tool
Version: 6.3.9600.17031

Applying image
[=====100.0%=====]
The operation completed successfully.
**** Mounting F:\Windows\System32\config\SYSTEM
****          to HKEY_USERS\Galileo-9548-SYSTEM
**** Setting hostname to Kowalski
**** Restoring time zone to 'Central European Standard Time'
****
**** Successfully applied d:\SW\9600.16384.x86fre.winblue_rtm_iotbuild.150309-0310_galileo_v2.wim
****          to F:\
****
****          hostname: Kowalski
****          timezone: Pacific Standard Time
****          Username: Administrator
****          Password: Quark
**** Done.

d:\SW>
```

Rys.11. End of the procedure of preparing the SD card with Windows 8.1.

**Note 2!**

When you create a bootable SD card script *apply-bootmedia.cmd* on a PC changes the time zone. Make sure that after this procedure, the time setting will return to the original time zone.

**Note 3!**

Make sure that host name and password was correctly written down.

Insert the prepared SD card in the slot on the Intel Galileo board and connect the power cord. On your PC, launch the Galileo Watcher application. The shortcut to this application should be on your Windows desktop. Connect the power and wait for the launch of Windows 8.1 on Intel Galileo. During the startup process "SD" LED diode placed next to the USB host port should be blinking.

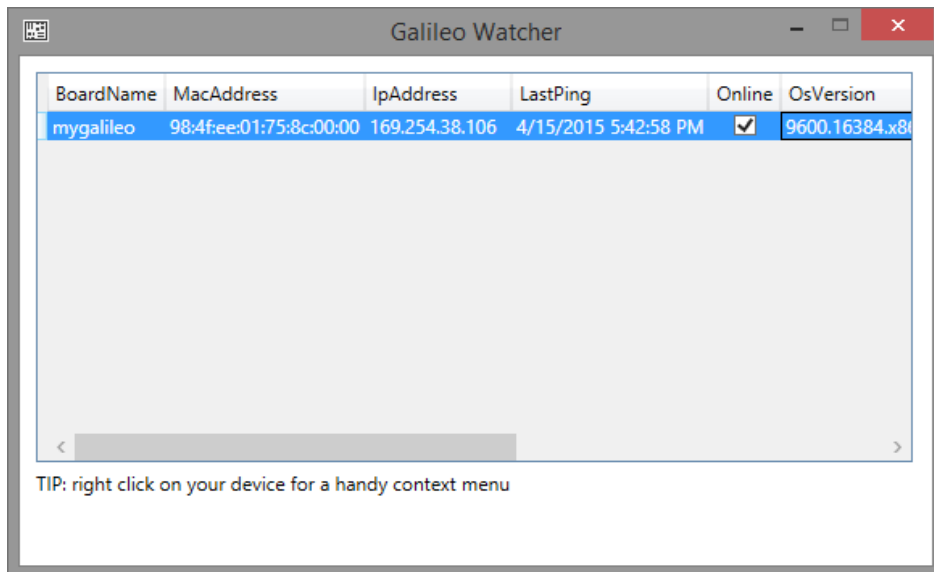


Fig.12. The main window of the Galileo Watcher application.

After several seconds, the Galileo Watcher application should detect a new host in local area network (new Intel Galileo board) with the name specified when creating a bootable SD card (Fig. 12). In the window of application, there are several information such as a host name, MAC address of network wired adapter and an assigned IP address.

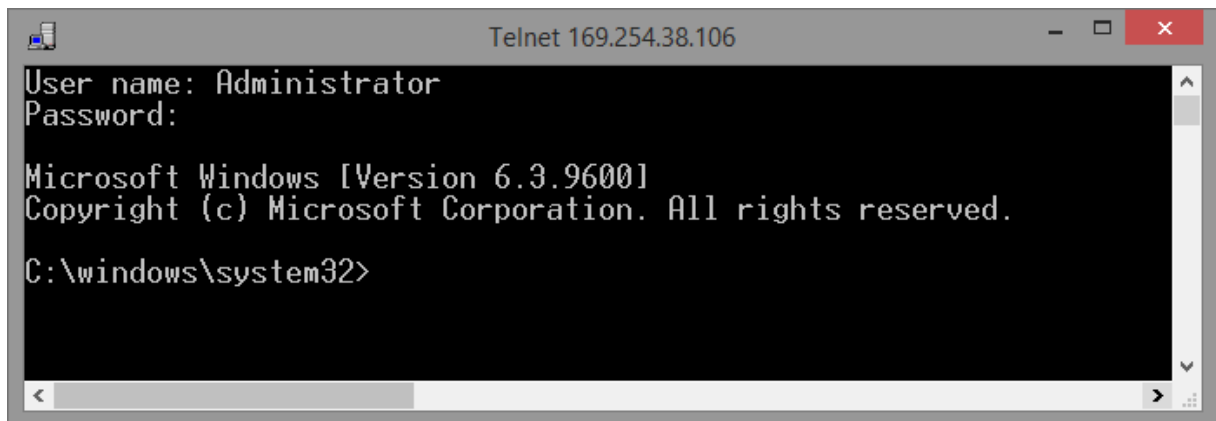


Fig.13. Logging window to Windows 8.1 on Intel Galileo.



By using the Telnet protocol through an option “Telnet Here” connect with your own Galileo board by clicking right mouse button on it. After entering the **Administrator** username and password provided in the preparation of the SD card you will be logged into the operating system on the Galileo board (Fig. 13).

The correct of installation of a wireless network card driver Intel® Centriano® Wireless-N 135 can be checked by searching all devices running on a Galileo platform issuing the command:

```
devcon find *
```

Finding on the displayed list wireless network card, we learn that it has a device ID as shown in Figure 14.

```
Telnet 169.254.38.106
ACPI\PNP0000\4&431F7F5&0 : Programmable interrupt controller
PCI\VEN_8086&DEV_0937&SUBSYS_09378086&REV_10\3&B1BFB68&0&A6 : Intel Quark STMAC PCI Ethernet Adapter #2
PCI\VEN_8086&DEV_0937&SUBSYS_09378086&REV_10\3&B1BFB68&0&A7 : Intel Quark STMAC PCI Ethernet Adapter #3
ROOT\KDNIC\0000 : Microsoft Kernel Debug Network Adapter #1
ACPI\INT3499\2&DABA3FF&0
PCI\VEN_8086&DEV_0934&SUBSYS_09348086&REV_10\3&B1BFB68&0&AA : DMap Direct Memory Mapper Device
PCI\VEN_8086&DEV_0892&SUBSYS_00628086&REV_C4\4&6952C7A&0&00B8: Intel(R) Centriano(R) Wireless-N 135 #4
USB\ROOT_HUB\4&21B6B18C&0 : USB Root Hub
USB\ROOT_HUB20\4&30CFB162&0 : USB Root Hub
ROOT\ACPI_HAL\0000 : ACPI x86-based PC
```

Fig.14. List of devices operating on the Intel Galileo under Windows 8.1.

Knowing the hardware identifier, you can get information about the status of the installed network card driver by typing:

```
devcon status *DEV_0892*
```

The message from Figure 15 shows that installation of the drivers for the network card was correct. Otherwise, the command: **devcon status \*DEV\_0892\*** would display an error message 28, which means no installed drivers.

```
Telnet 169.254.38.106
C:\windows\system32>devcon status *DEV_0892*
PCI\VEN_8086&DEV_0892&SUBSYS_00628086&REV_C4\4&6952C7A&0&00B8
Name: Intel(R) Centriano(R) Wireless-N 135 #4
Driver is running.
1 matching device(s) found.
C:\windows\system32>
```

Fig.15. Status of the wireless network adapter Intel® Centriano® Wireless-N 135 drivers

Programming of Intel Galileo platform running under the Windows operating system 8.1 is done using Visual Studio Express 2013 for Windows Desktop.

Start Visual Studio Express 2013 for Windows Desktop. From the menu, select *Plik->Nowy projekt*, and then in the window as shown in Figure 16, the template of Visual C++ application, *Windows for IoT*. Name the project and click OK.

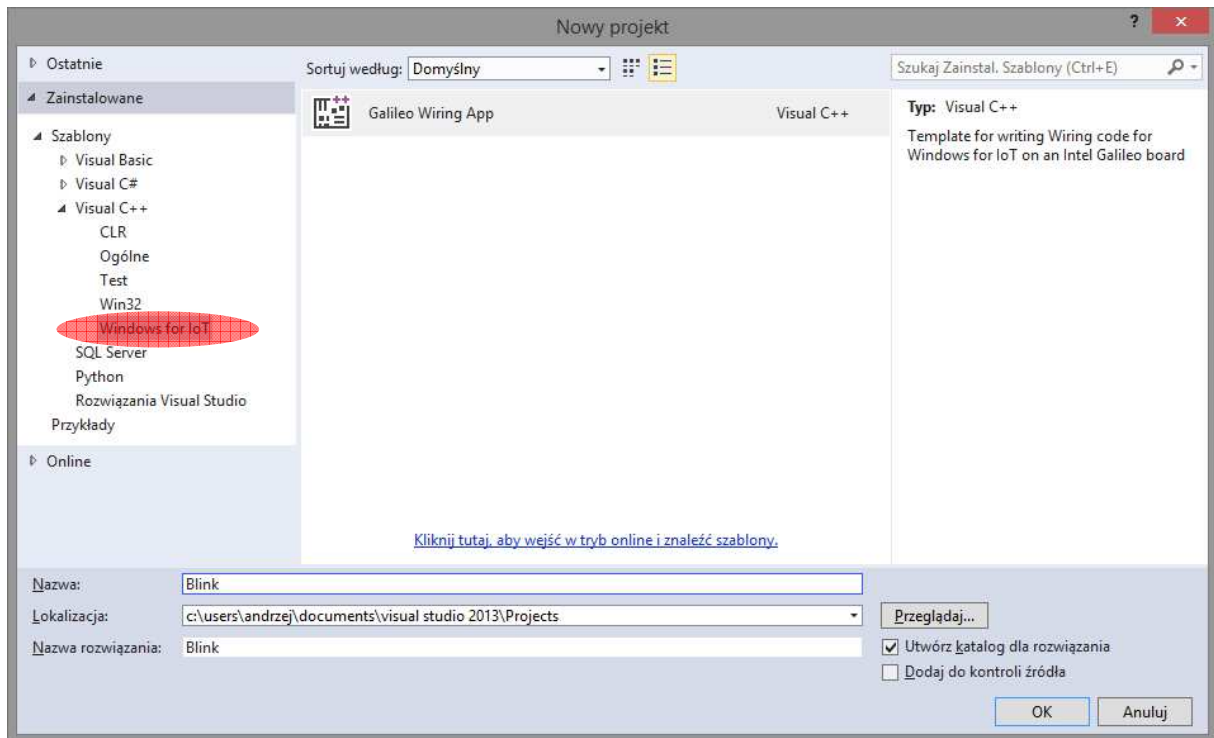


Fig.16. The Nowy projekt window of Visual Studio Express 2013 for Windows Desktop

By default each new project is filled with Blink program code known from the Arduino IDE. Its task is to alternate lighting and blanking the LED "L" next to the USB host port on the board of Intel Galileo.

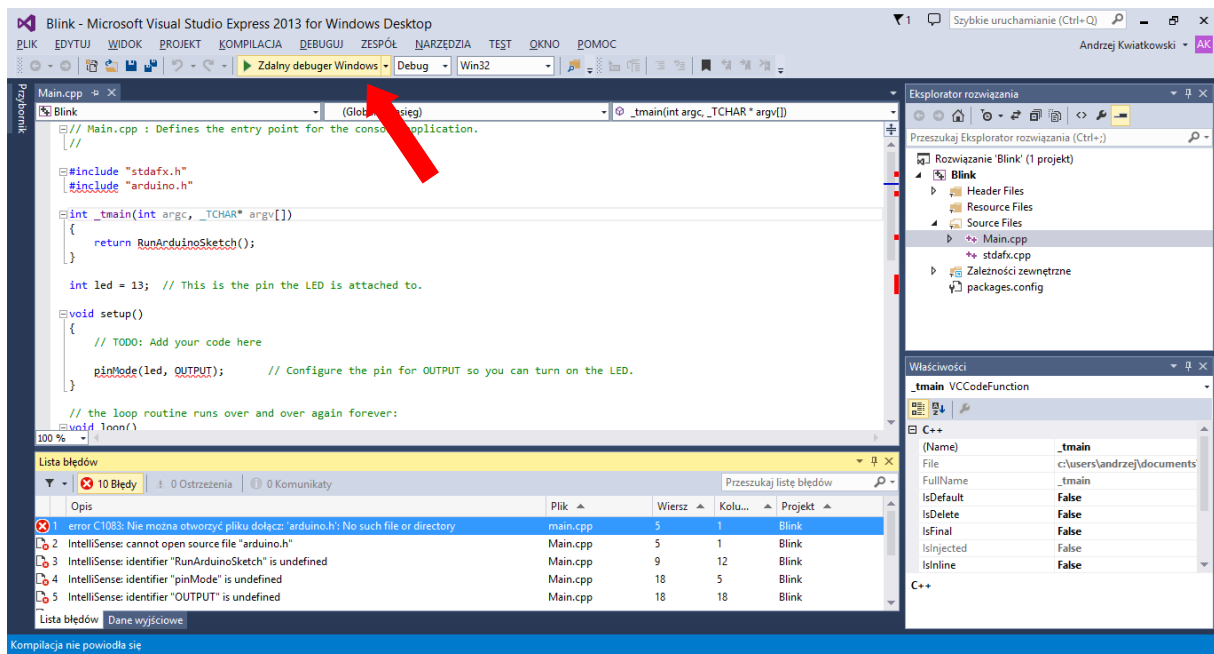


Fig.17. The main window of Visual Studio Express 2013 for Windows Desktop

Compiling and implementation an application follow after clicking on the button: "Zdalny debugger Windows" (Fig.17.). Sometimes the environment does not find the Arduino platform libraries and the appropriate functions declarations from the Arduino IDE.

Then in the menu **Narzędzia->Menadżer pakietów NuGet->Zarządzaj pakietami NuGet** for solutions the appropriate package files should be attached to the project. In this case it will be **Microsoft IoT C++ SDK** package available from 2 December 2014 (Fig.18.).

Release package date is important because the previous version of this package called Galileo C++ SDK is no longer compatible with the latest version of the system Windows 8.1.

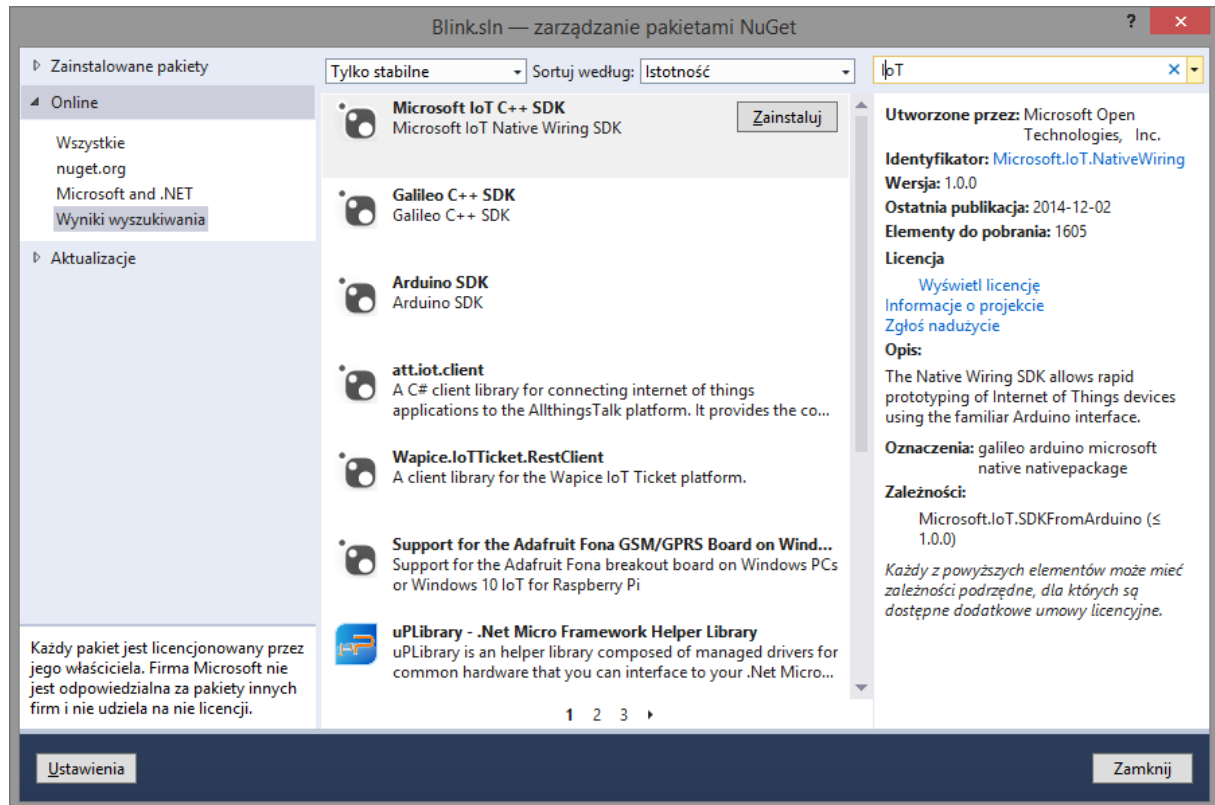


Fig.18 Adding packages Nuget Visual Studio Express 2013 for Windows Desktop

After attaching the necessary packages the project properties must still be changed. In the field **Właściwości konfiguracji->Debugowanie** the name of the remote server should be changed for the hostname given on the stage of creating a bootable SD card (Fig.19).

After changing the properties and start the compilation stage of implementation (project transfer to a Galileo board) begins with entering user name and password provided at the stage of creating the SD card.

Programming of Intel Galileo board under Windows 8.1 controll in Visual Studio Express 2013 for Windows Desktop is done using the syntax of the language used in the Arduino IDE. Each compilation of new applications write on the SD card EXE file (Fig.20) in the folder specified in the project properties. This file can be run from the Galileo platform level. Access to this folder can be achieved by right-clicking on the Intel Galileo board detected in the Galileo Watcher application and selecting **Open Network Share** position.

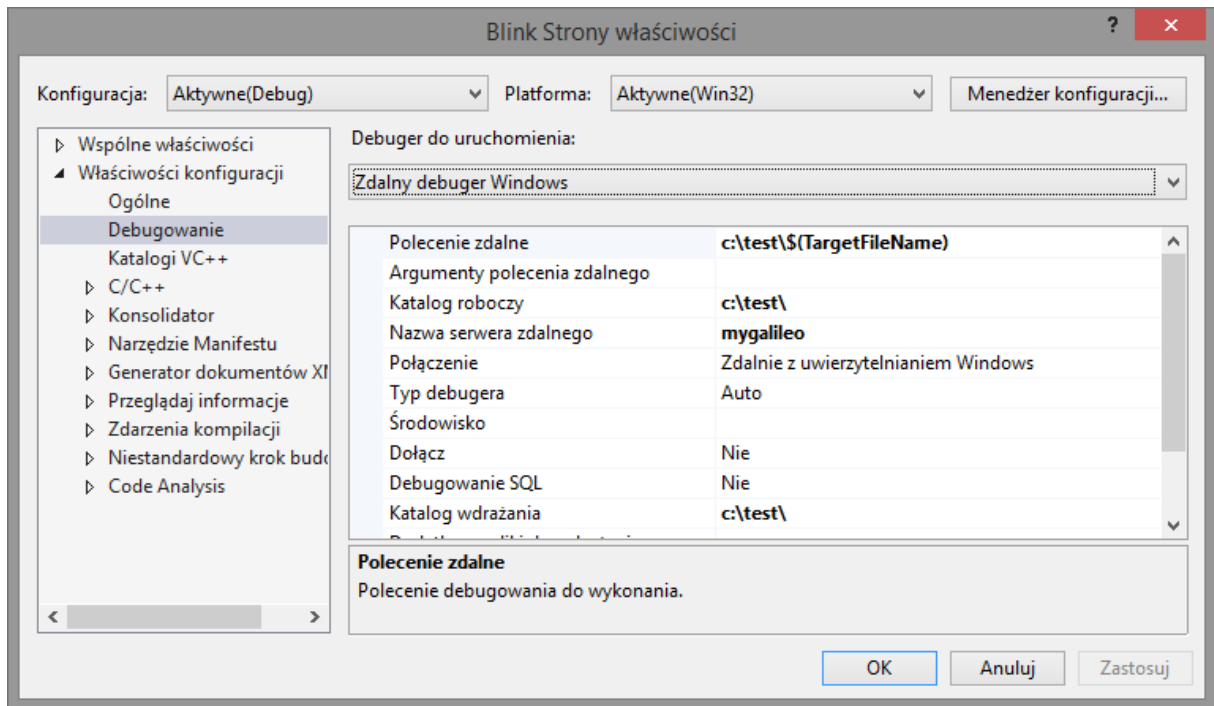


Fig.19. Project Properties window of Visual Studio Express 2013 for Windows Desktop

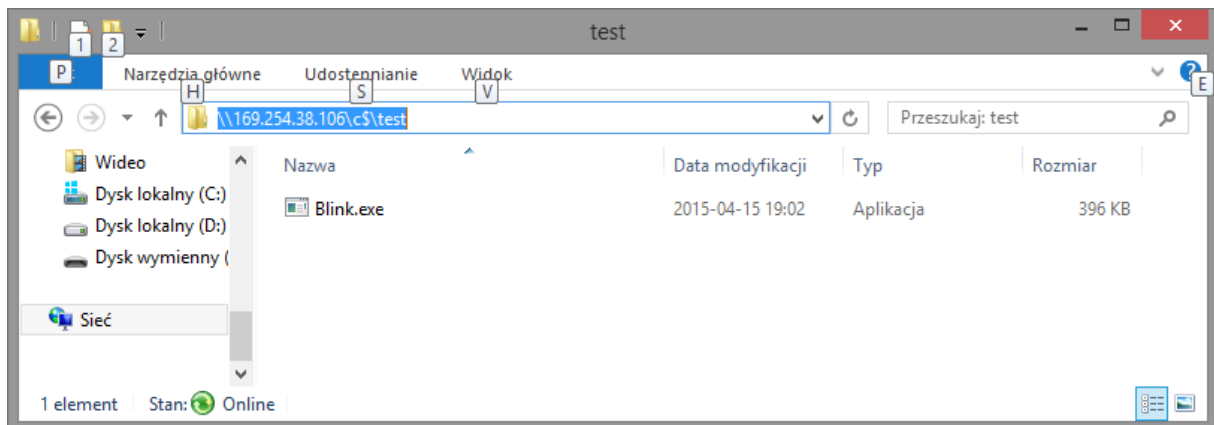


Fig.20. The folder containing the executable files on Intel Galileo under Windows 8.1 control.

Shutting down the Windows system working on Intel Galileo board follows after issuing in the Telnet window command:

```
shutdown /s /t 0
```

After the „SD” diode is not more flashing power the power can be turn off. Disconnecting the power without properly shutting down Windows will greatly extending the launch of the system on reboot, which is associated with searching for errors on the disk (SD card) just as happens in the case of PCs.

### 3. End tasks

1. Using the sound sensor and a relay from the Grove Starter Kit Plus write a program, which realizes the function of the switch reacting to clap hands. The number of switching the transmitter display on the LCD screen.
2. Write a program of twilight sensor (Light sensor), whose sensitivity is adjustable via a potentiometer connected to the port A0 on Base Shields board of the Grove Starter Kit Plus.

During the final task, you can use the example programs (ie. Sketches) supplied by the producer of Grove Starter Kit Plus Intel IoT Edition, for the Arduino IDE. The website: <https://ms-iot.github.io/content/SampleApps.htm> can be helpful as well.