GSoC Application / KDevelop Integration with Arduino and embedded development

Patrick José Pereira

Electronic engineering, Federal University of Santa Catarina, Brazil, e-mail: patrickelectric@gmail.com, tel: +55-48-9917-4777

1 Motivation

The actual Arduino IDE, was initially created with Java[1] and is still a simple IDE[2] that does not provides autocompletion, sentence errors, assembly visualizer, field for compiling and linking flags, syntax highlighting and other features present in both Kdevelop and Qt Creator.
The other alternative IDEs for Arduino are:

- PROGRAMINO IDE
  Price: 30€
  It provides good features and development options, but isn't a free or open source software.
- PlatformIO IDE
  Price: FREE
  A good substitute to Arduino IDE. It's a modified Atom version, Doesn't have assembly visualization, serial plotting, data loggging and other important tools. It's written in CoffeeScript, JavaScript, Less and HTML.
- Embrio
  Price: $50
  An easy variable visualization ambient with "real-time" plot with code editor. But, isn't free or open source.

Some developers and educators say that Arduino IDE isn't as good and comfortable as a development ambient for higher education and development. In order to the to fix this situations, the development of Arduino plugins for KDevelop, Qt creator, Visual Studio and Eclipse began, but generally the setup for such plugins is complicated and it's necessary a good understanding of these IDEs and OS's ambient functionality.

2 Project goals

Here I will showcase some of the features that I propose to implement. Right now I'm the second biggest contributor of a project called ArduIDE, a C++ IDE to help with software development to Arduino boards. The initial idea is to work with ArduIDE to port some implemented features that already exist in Qt4 to KDevelop in Qt5 and some news features like: memory map view, assembly visualizer and others.

Some of the goals are:

1. Port ArduIDE features from Qt4 to Qt5 into KDevelop. (on going)
2. Port Arduino support from 1.6.0 to last Arduino version into KDevelop.

[1] ARDUINO 0001 at 08/25/2005
[2] Integrated development environment

3. Port and create others features like:
    - Board selector.
    - Clock selector.
    - Serial monitor.
    - Assembly visualization.
    - Compilation flags editor.
    - Real-time plotting with serial input, with integrator and derivator plot.
    - Real-time data logging.
4. Perform the upload process for Arduino boards with AVR processors.
5. Perform the upload process for Arduino boards with ARM processors.
6. (If time isn't a problem) Realize the upload process for Arduino boards with x86 processors.
   With all Arduino features implemented, the development of ARM processors can start:
7. Implement same features of Arduino development with ARM processors.
8. Add support to JTAGs[3] with OpenOCD.
9. Add GDB debugger with OpenOCD interface.

[3] Joint Test Action Group

3 Timeline

This section shows the estimated timeline of the project.

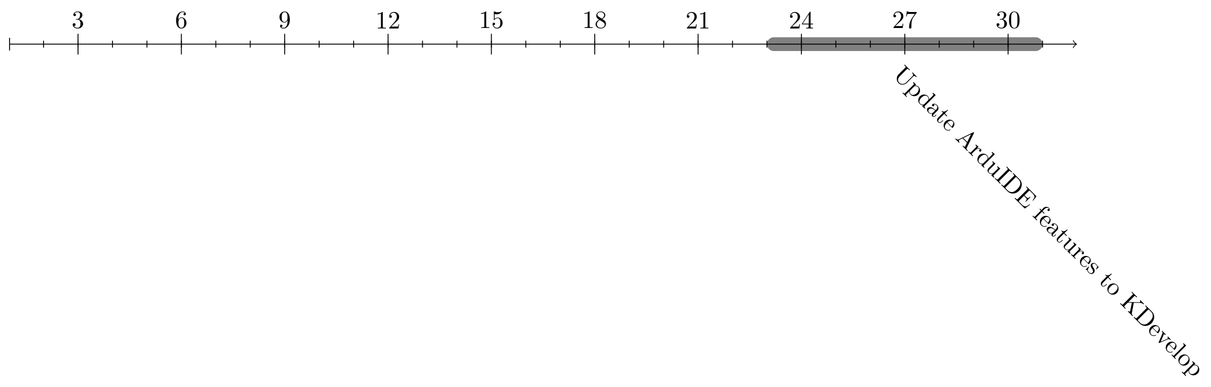
3.1 May:

Initial week, change ArduIDE features from Qt4 to Qt5 in KDevelop. Almost everything in ArduIDE is done with Qt4, Grantlee and QScintilla

It's possible to replace grantlee with QML since ArduIDE create only a dynamic webpage with examples and libraries installed on the system. Right now a good part of ArduIDE it's in Qt5 in my personal repository but isn't finished yet.

Some of the features that already exist in ArduIDE that will be ported to KDvelop:

- Board selector
- Clock selector
- Serial monitor
- Assembly visualization
- Compilation flags



3.2 June:

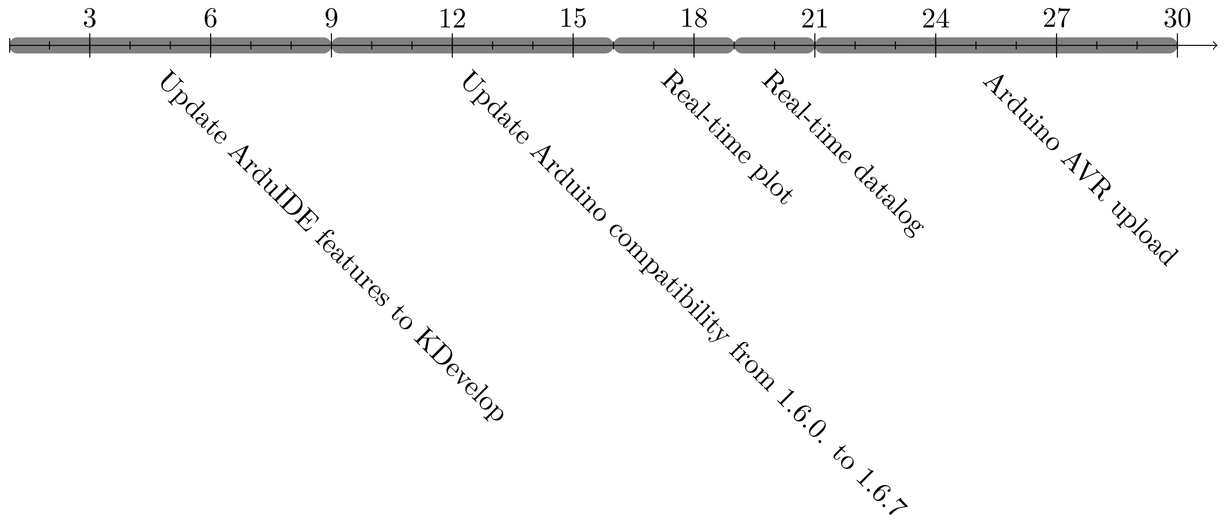
After the transition from Qt4 to Qt5, the IDE needs to be updated to the last version of Arduino software and be compatible with the boards[4].

Test the software compatibility and programmability of KDevelop with Arduino's source code and libraries. After that, a plugin[5] will be created to manage what kind of embedded system the project will use during the development, this same plugin will manage other plugins that will provide a variable system development.

In the beginning will be only one children plugin to manage the Arduino environment and process. To perform such boards and processors management it's possible to use a generic makefile[6] to compile and upload the project to Arduino boards.

With selectors menu, like Board and processor clock, it's possible to use the Kdevelop module called Sublime with a drop-down menu in toolbar that will be display the options to the user, and with this information it's possible to modify a generic Arduino makefile to manage the project compiling and upload process.

The data-log can be added in a Sublime Dock, the same back-end of data-log can be used to the a plot system[7] that will provide in the same Dock a user visualization tool to display the output data of the board.



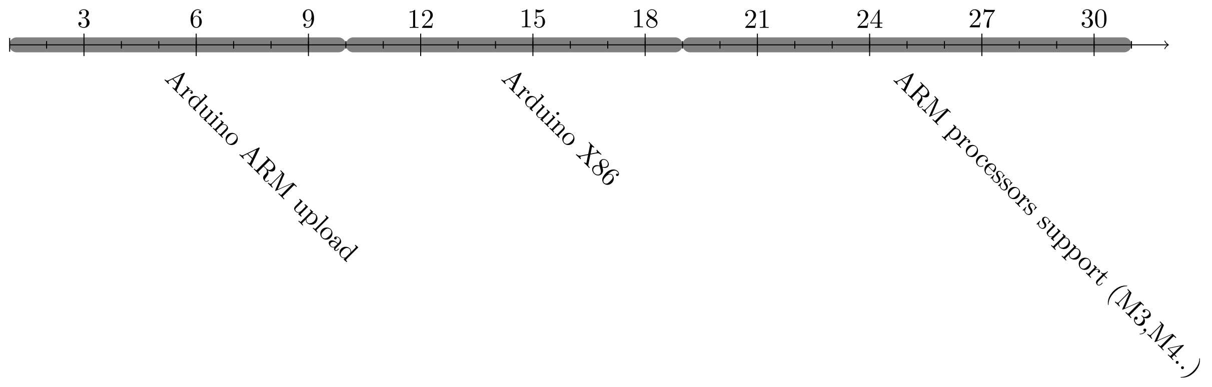
[4]Until now I only have an Arduino Nano, Mini, Uno and access to an ARM version.
[5]Plugin name: Embedded system development
[6]An example to this can be saw in arduino-makefile and in mididuino
[7]In Arduino IDE 1.6.6 it was added a serial plotter.

3.3 July:

With all Arduino AVR boards working, the process of Arduino ARM boards begun. Unlike the AVR boards which use avrdude, the ARM boars use bossac to perform the upload process. The same process of Arduino AVR can be used with the Arduino ARM boards, using a generic makefile, but with bossac and arm-none-eabi-binutils[8] in the place of avrdude.

The Arduino X86 boards uses a ~~complicated, obscure and catastrophic hacked~~ script to perform the compile and upload process. If time isn't a problem until now in the project the implementation of such feature will be performed.

With arm-none-eabi-binutils we already have a .bin, .elf or .hex to be uploaded to an ARM processor. It's necessary something between the processor and the computer to realize the upload process, this thing it's called JTAG or programmer interface, can it be a Pirate BUS, JTAG, DTAG and ICSP interface or a simple FTDI232R/H to realize the midfield to the in-chip JTAG. It's possible to communicate with all this interfaces with OpenOCD, a list with all possible interfaces can be located here. Using OpenOCD can be a bit complicated but not impossible !

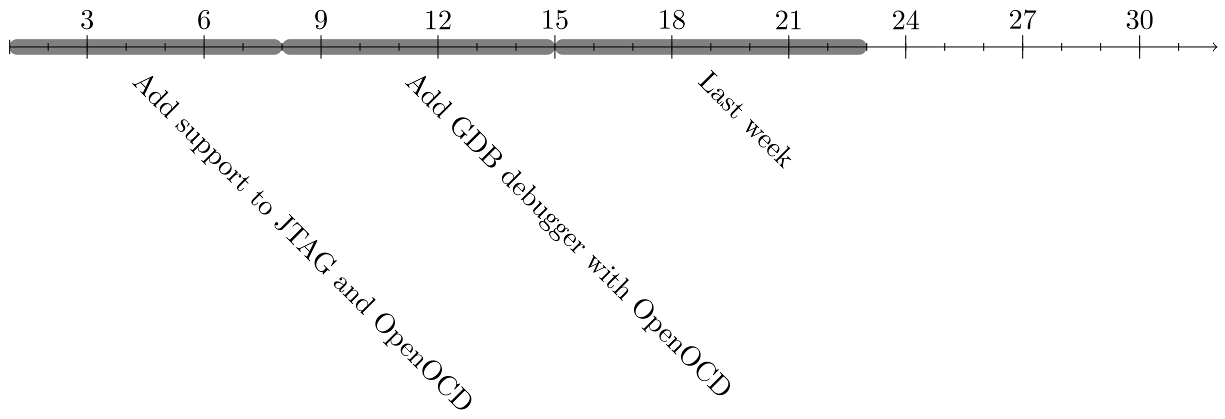


[8] Cross compiler for ARM EABI (bare-metal)

3.4 August:

If the last point finish with success, this part can be done without so much problems. With a OpenOCD link with the board interface, it's possible to use the GDB with a remote access to the socket door that OpenOCD open to realize the communication with GDB and the hardware components of the processor debug.

With all this steps done, we have created a awesome embedded system platform with our friend KDevelop :)
The last week will be reserved to solve some bugs and finish some points in the documentation of the project.



4 Me

My name is Patrick José Pereira, from: Florianópolis, Santa Catarina, Brazil. Studied industrial robotics in SENAI for 2 years and right now on the last year of electronic engineering and realizing an internship at Intel Semiconductor in the OTC[9].
Since at beginning of 2012 I am a member of ROBOTA[10] and until the end of 2015 member of proVANT[11].
I realize a lot of important things in my live and until now, and I am very happy with my internship at Intel (a childhood dream). I moved out from my city and right now I am living in campinas with a crazy guy that invite me to contribute with some KDE project and here I am.

I have experience with C, C++, Python, JS and other programming languages. I have already worked with AVR, ARM, embedded linux, embedded Real-Time operating systems (FreeRTOS), some development boards like: Beaglebone, Raspberry, Discovery Board STMF32M4, STELLARIS Development Board with a Cortex M4.
One of the things that I love is the integration of hardware, software and engineering to realize my objectives: Radio demodulation, online FFT to help with filter project, a simple GroundCountrol and data-log using QT and python and other things that I don't know where they are.

[9]Open Source Technology Center
[10]Robotic Developer and Competition Group
[11]Research and Development of Unnamed Aerial Vehicles

NOTE: I only cited ArduIDE because it's a project that I work some times, it's only for an example.It's almost the same thing that saying that I want to port some base features of any embedded IDE envoryment.

The idea is to porte some of existing features that an embedded program must have.