## 1.1 Потребни елементи за користење на плочката

За да може да се користат можностите на Olimex LPC-P2148 плочката потребни се [http://www.olimex.com/dev/pdf/ARM/LPC/LPC-P2148.pdf]:

* Кабли – зависат од изборот на дебагерот кој го користиме и тоа за:
  + ARM-USB-ODC - потребен е 1,8 метарски A-B USB кабел заедно со RS232/JTAG кабел
  + ARM-USB-TINY - потребен е само 1,8 метарски A-B USB кабел
  + ARM-USB-JTAG - потребен е LTP кабел
* Хардвер – потребен е Olimex дебагер и тоа:
  + ARM-USB-ODC [http://www.olimex.com/dev/arm-usb-ocd.html]
  + ARM-USB-TINY [http://www.olimex.com/dev/arm-usb-tiny.html]
  + ARM-USB-JTAG [http://www.olimex.com/dev/arm-jtag.html]
* Софтвер – потребен е ARM C компајлер заедно со JTAG програмер. Препорачливо е да се користи интегрирана развојна околина за програмирање поради едноставноста за користење и напредните можности кои ги нуди при отстранување на грешки
  + Комерцијална развојна околина – Crossworks for ARM – напредна развојна околина со поддршка за развој на C/C++ и асемблерски код за ARM7, ARM9, Xscale и Cortex-M3 микроконтролерите [http://www.rowley.co.uk/arm/]
  + Бесплатна развојна околина – Eclipse со GNU C компајлер и OpenOCD

Пакетот кој ние го користиме за тестирање на можностите на Olimex LPC-P2148 развојната плочка покрај самата плочка содржи и ARM-USB-ODC програмер/дебагер, JTAG кабел (20cm), кабел за напојување (30cm) како и инсталациски диск со потребните драјвери за дебагерот и бесплатна алтернатива на комерцијалните развојни околини.

## 1.2 Опис и инсталирање на потребните елементи

ARM-USB-ODC комбинира JTAG хардвер за дебагирање, RS232 виртуелна порта и адаптер за напојување во еден уникатен и компактен уред со димензии 50x40 mm. Новите преносни компјутери немаат сериски или пак паралелни порти, па овој уред е погоден за користење бидејќи има USB 2.0 порта за лесно поврзување и дебагирање сите ARM уреди кои се поддржани од OpenOCD.

Овој дебагер ја користи стандардната 2х10 JTAG порта на ARM микропроцесорот, создава виртуелна RS232 порта на компјутерот преку кој се врши програмирањето со сите поддржани команди, може да напојува процесори кои имаат работен напон од 2.0V – 5.0V додека пак со преместување на џамперите на самиот уред целата плочката може да се напојува со 5V, 9V или 12V (ако двата џампери се отворени плочката се напојува со 12V, ако десниот џампер е затворен плочката се напојува со 9V, а ако левиот е затворен тогаш плочката се напојува со 5V).



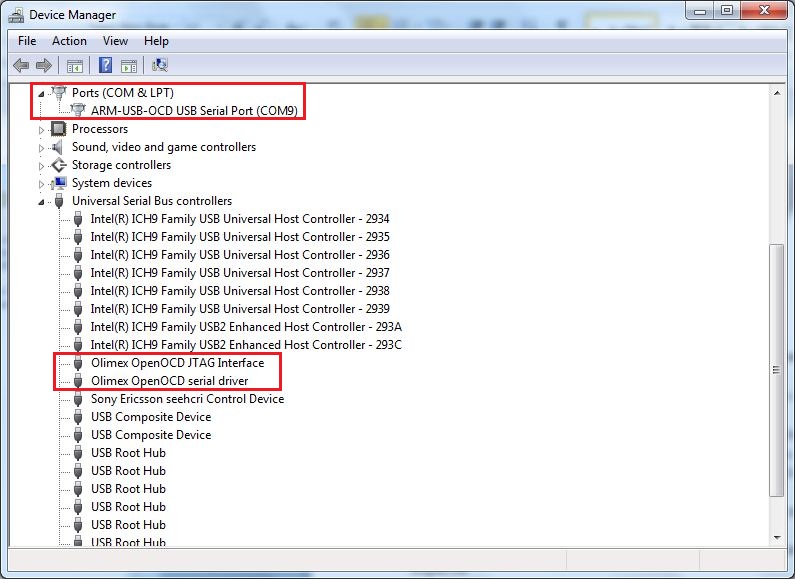
За да може да се користи овој уред потребно е да се инсталираат потребните драјвери за Windows оперативниот систем кои се вклучени во пакетот и се наоѓаат на инсталацискиот диск. Откако уредот ќе се приклучи на USB 2.0 портата преку USB A-B кабелот (Слика Х) системот ќе го регистрира уредот и неуспешно ќе проба да најде соодветни драјвери.



Истите треба рачно да се инсталираат на следниот начин (упатството е за Windows 7):

* се отвара Device Manager
* се избира уредот кој не е препознаен (жолт знак со извичник)
* со десен клик се избира Properties
* во Driver јазичето се избира опцијата Update Driver
* се избира Browse my computer for driver software
* како локација се специфицира ARM-USB-OCD-DRIVER папката која се наоѓа на инсталацискиот диск
* со кликнување Next системот ќе започне со инсталирање на соодветниот драјвер
* Постапката МОРА да се повтори три пати

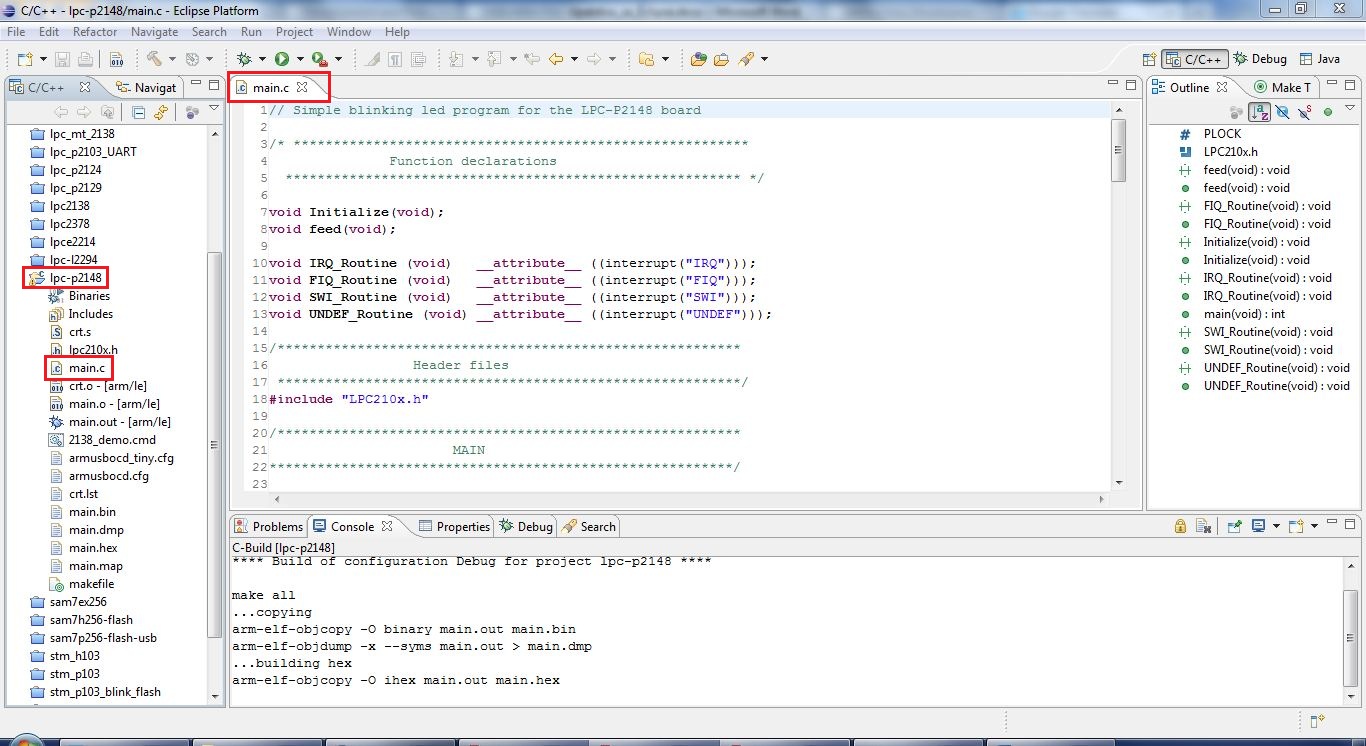
Само откако системот ќе го инсталира драјверот успешно три пати уредот може да се користи. На Слика Х е прикажана успешна инсталација на драјверите. Поврзувањето на каблите е тривијално. Ако дебагерот е поврзан со компјутер преку USB порта, a со плочката преку кабелот за напојување ќе светне соодветната диода за напојување (PWR).



Изборот на развојна околина е важна одлука која може значително да го олесни програмирањето и искористувањето на модулите кои ги нуди Olimex LPC-P2148 развојната плочка. Најпозната комерцијална развојна платформа е Crossworks for ARM, која е многу лесна за користење за дебагирање, но за нејзино користење по првите 30 дена наменети за евалуација мора да се купи лиценца. Моменталната цена за лиценца за персонална употреба изнесува 150 американски долари [http://www.rowley.co.uk/crossworks/Licensing.htm].

Како бесплатна алтернатива на комерцијалните платформи за развој на вградлив софтер ние ќе ја искористиме Eclipse развојната околина со потребните додатоци. Развојот на код, отстранувањето на грешки и менаџирањето со проекти е речиси идентично лесно, со тоа што инсталирањето и поставувањето на параметрите на развојната платформа е значително потешко. Постапката е добро документирана [http://gnuarm.alexthegeek.com/ARM\_Cross\_Development\_With\_Eclipse\_Components\_Revision3\_March\_17\_2006.pdf] и вклучена во инсталацискиот диск како извршна датотека која ги инсталира сите потребни надворешна алатки за функционирање на развојната околина:

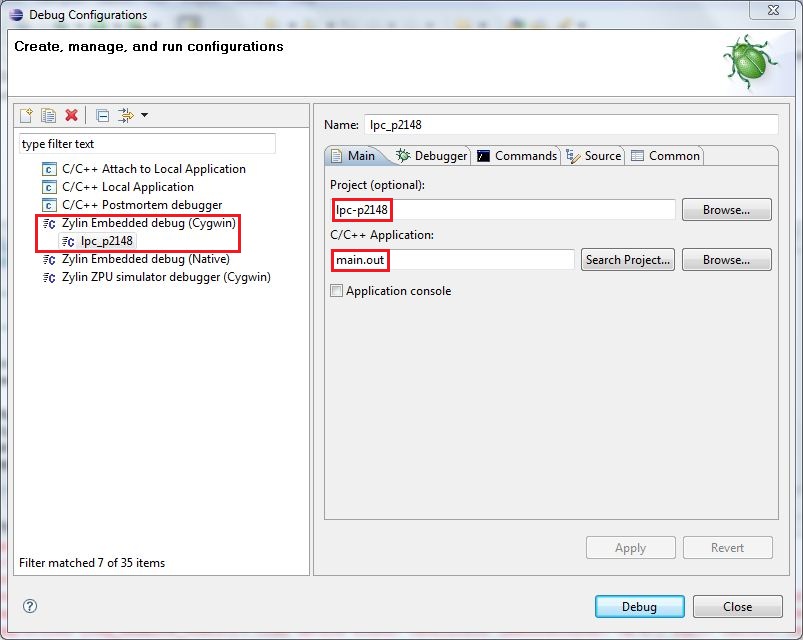
* Се започнува со извршување на Olimex-GCCFD-20101015.exe датотеката.
* Се следење на упатството се инсталира програмата.
  + Бидејќи потребно е инсталирање на GNU C компајлер (Unix\Linux) препорачливо е патеката каде што се инсталира програмата да биде без празни места или да се остави предефинираната.
* Откако инсталацијата ќе заврши, Eclipse развојната околина може да се подигне преку новосоздадената кратенка на работната површина.
* Во работната околина ќе бидат вклучени повеќе проекти за различни развојни плочки. Во нашиот случај го отвараме проектот именуван како lpc-p2148 и со двоен клик ја отвараме main.c датотеката. Овој чекор е прикажан на Слика Х.



* Со Project -> Build All го компајлираме проектот при што се создава main.out излезна датотека.
* Се префрламе во Debug перспектива (main.c мора да биде отворена).
* Преку Run -> External Tools -> OpenOCD го активираме чип дебагерот
  + Во конзолата ќе се испишат информации со црвена боја.
  + Ако се појави грешка како на Слика Х , тоа значи дека main.c датотеката не е селектирана во Debug перспективата.



* Преку Run -> Debug Configurations треба да направиме профил за дебагирање
  + Се избира Zylin Embedded debug (Cygwin) и со New се прави нов профил
  + Во Main јазичето го избираме посакуваниот проект и C/C++ апликацијата (main.out) како што е прикажано на Слика Х.



* + Во Debugger јазичето во GDB debugger полето се поставува вредноста arm-elf-gdb.exe како што е прикажано на Слика Х.
  + Со кликнување на Debug копчето започува пренесувањето на програмата во флеш меморијата на развојната плочка
* Во конзолата ќе се испишат пораки за грешки и навидум нема ништо да се случува на плочката.
* Овој проблем се решава со вадење на кабелот за напојување од дебагерот. При повторното приклучување на истот кабел програмата која сме ја пренеле во меморијата на плочката ќе почне да се извршува (однесувањето на плочката зависи од модулите и функионалностите кои сме ги предвиделе во кодот).
* Создавање на нов проект најлесно се реализира со копирање на lpc-p2148 проектот и менување само на main.c дадотеката при што може да се користи постоечката конфигурација за дебагирање.

