

# Introduktion till Atmel Studio

## 1 Inledning

Det här dokumentet syftar till att ge grundläggande kunskaper om hur man använder Atmel Studio. Fokus ligger på att kunna använda denna utvecklingsmiljö för programmering av enchipsdatorn på Arduino Leonardo (ATmega32U4). Eftersom utvecklingsmiljön endast kan köras i Windows så förutsätts det att man har tillgång till en dator med detta operativsystem.

Detta dokument berör följande:

- Installation av drivrutiner och utvecklingsmiljö.
- Skapa projekt för programmering i Assembler eller C.
- Kompilering och överföring av program till enchipsdatorn.

## 2 Installation av drivrutiner och utvecklingsmiljö



Figur 2-1: Kontroll av enhetsbeteckning

Drivrutinerna och utvecklingsmiljön bör redan vara installerade på datorerna i labbsalarna. Men om du använder en egen dator så måste du utföra följande instruktioner.

### 2.1 Installation av drivrutiner

För att din dator ska kunna kommunicera med Arduino Leonardo behövs drivrutiner för detta. Tyvärr blir du tvungen att även ladda ner och installera Arduinos utvecklingsmiljö.

Besök följande hemsida (använd länken på It's Learning): <http://arduino.cc/en/Main/Software>

#### 1.2.1 Kontrollera enhetsbeteckning

Efter att drivrutinerna har blivit installerade bör du kontrollera vilken beteckning som tilldelats Arduino-kortet. Öppna *Device Manager* (Enhetshanteraren). När man strömsätter kortet, alternativt trycker på RESET-knappen (på Arduino-kortet), tilldelas den en viss beteckning, i detta fall COM11 (se Figur 2-1). Det är denna beteckning som ska användas när man för över program till enchipsdatorn.

*OBS! Denna beteckning är endast giltig under*

*ett kort ögonblick (cirka 10 sekunder)! Under denna tid väntar enchipsdatorns bootloader på en eventuell programöverföring. Kom ihåg följande när det är dags att föra över program:*

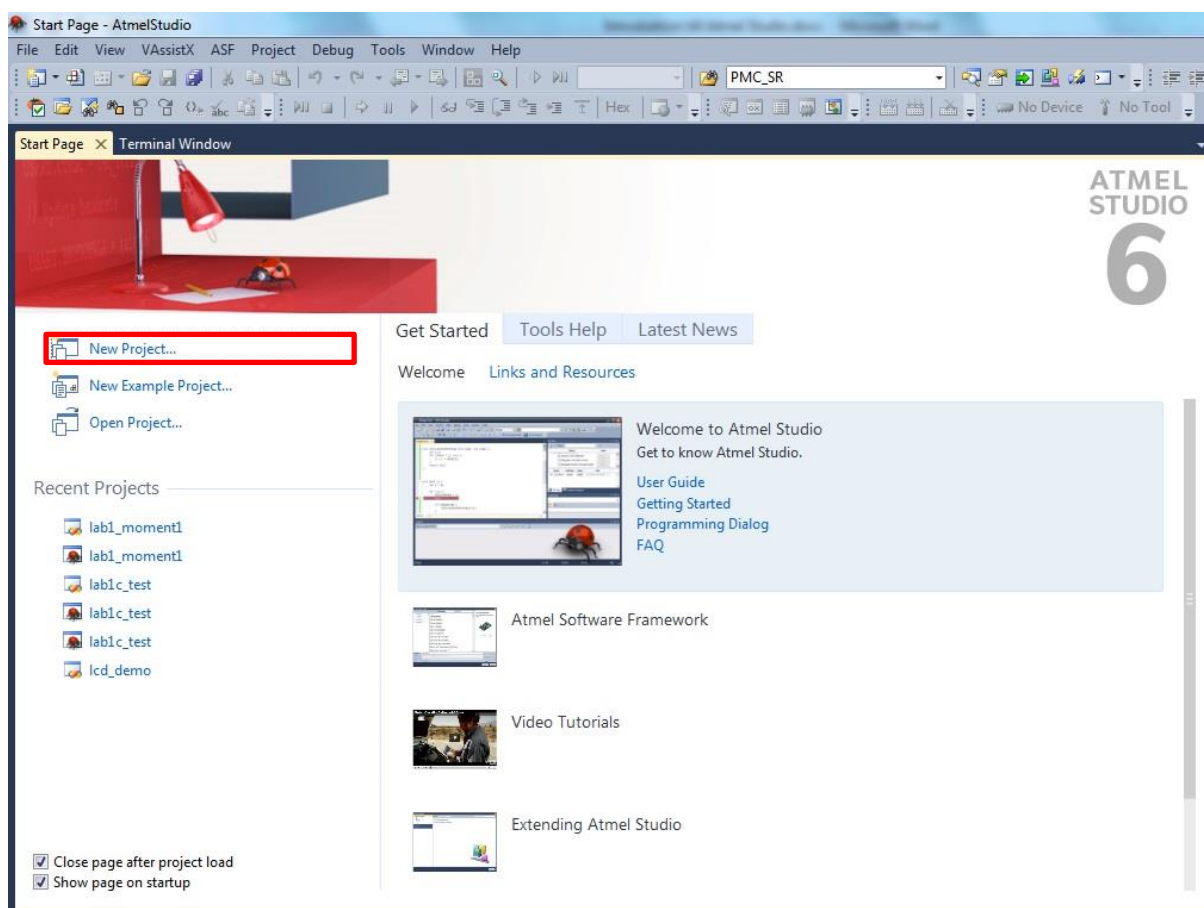
1. Förvissa dig om vilken enhetsbeteckning ditt Arduino-kort har.
2. Efter kompilering, tryck på RESET-knappen.
3. Vänta någon sekund. För sedan över programmet.

## 2.2 Installation av utvecklingsmiljö

Besök följande hemsida för att ladda ner den senaste versionen av Atmel Studio (använd länken på It's Learning): [http://www.atmel.com/microsite/atmel\\_studio6/default.aspx](http://www.atmel.com/microsite/atmel_studio6/default.aspx)

## 3 Skapa projekt för programmering i Assembler eller C

Efter att Atmel Studio har startat ska du klicka på *New Project* (se markering i Figur 3-1) alternativt klicka på menyn: *File -> New -> Project*.

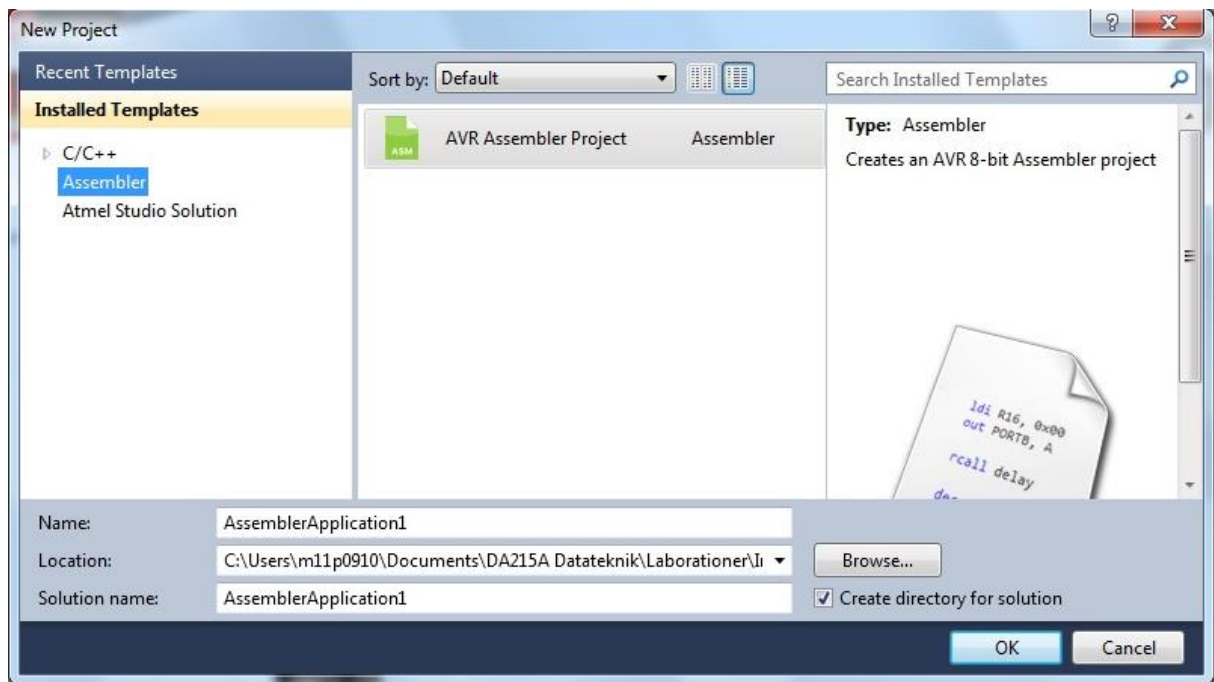


Figur 3-1: Startmeny i Atmel Studio

Nu är det dags att välja programmeringsspråket som är aktuellt för projektet.

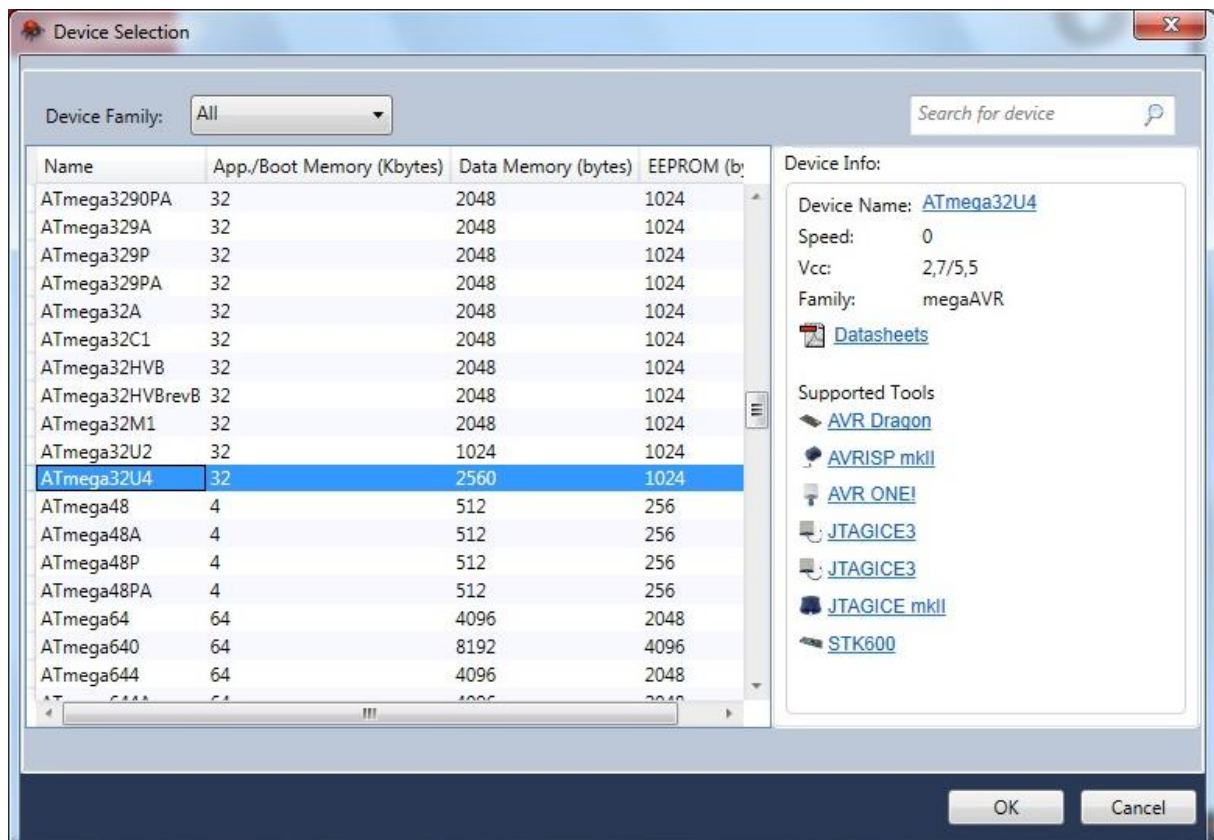
### 3.1 Assembler-projekt

Under *Installed Templates* finns alternativet *Assembler* (se Figur 3-2). Klicka på det. I rutan *Name* anger du vad projektet ska heta. Låt *Create directory for solution* vara ikryssad. Då kommer en mapp skapas med samma namn. Var denna mapp ska placeras väljer du genom att trycka på *Browse*. **OBS! Detta gäller även om du skapar ett projekt för C-programmering!**



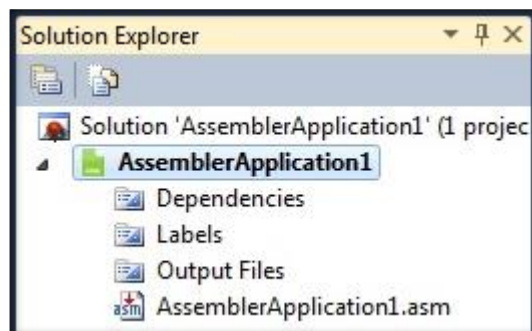
Figur 3-2: Inställningar för Assembler-projekt

Nästa steg blir att välja enchipsdatorn som programmet ska kompileras för (se Figur 3-3). I detta fall ska du välja ATmega32U4. *OBS! Detta gäller även om du skapar ett projekt för C-programmering!* Efter detta steg är det dags att börja programmera.



Figur 3-3: Val av enchipsdator

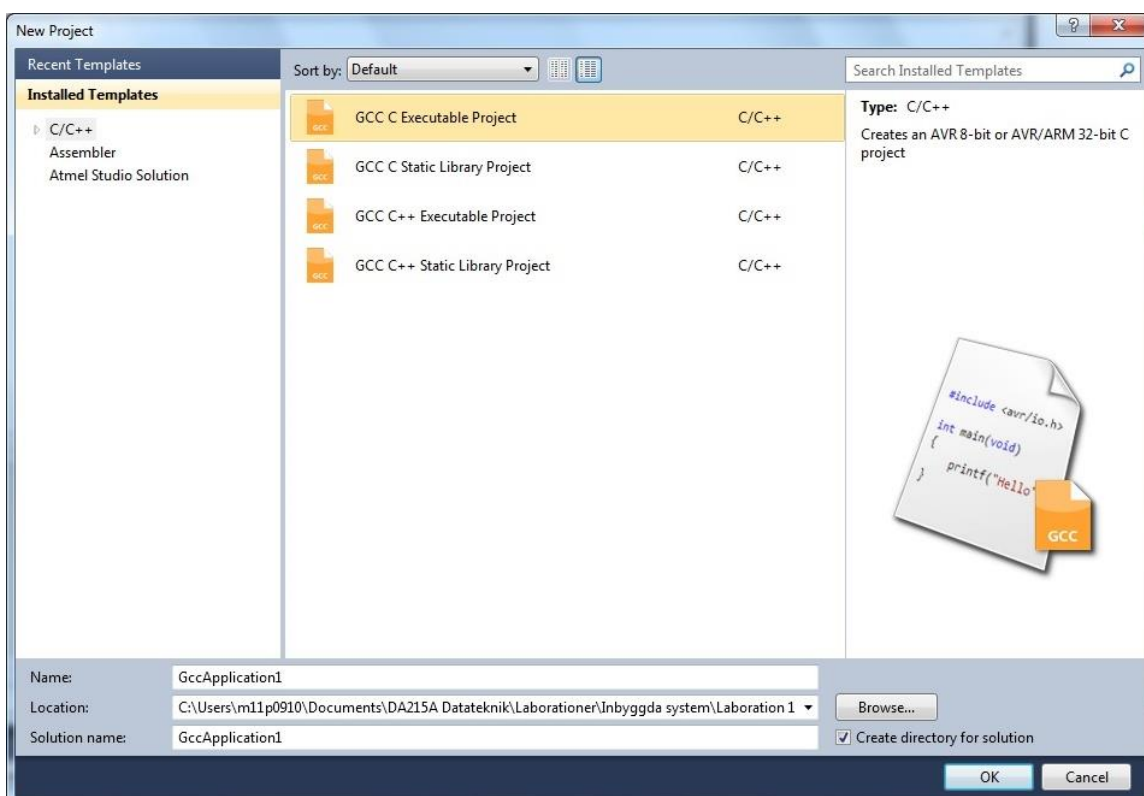
Till höger om fönstret för programkoden finns ett fönster som heter *Solution Explorer* (se Figur 3-4). Om man högerklickar på projektnamnet (*AssemblerApplication1* i detta fall) så får man fram en meny, som bland annat har funktioner för filhantering (lägga till filer, radera filer, skapa mappar, byta filnamn, etc).



Figur 3-4: Projektets struktur

## 3.2 C-projekt

Välj *C/C++* under *Installed Templates* och därefter *GCC C Executable Project* (se Figur 3-5). I övrigt gäller samma saker som beskrevs i det föregående kapitlet (3.1 Assembler-projekt).

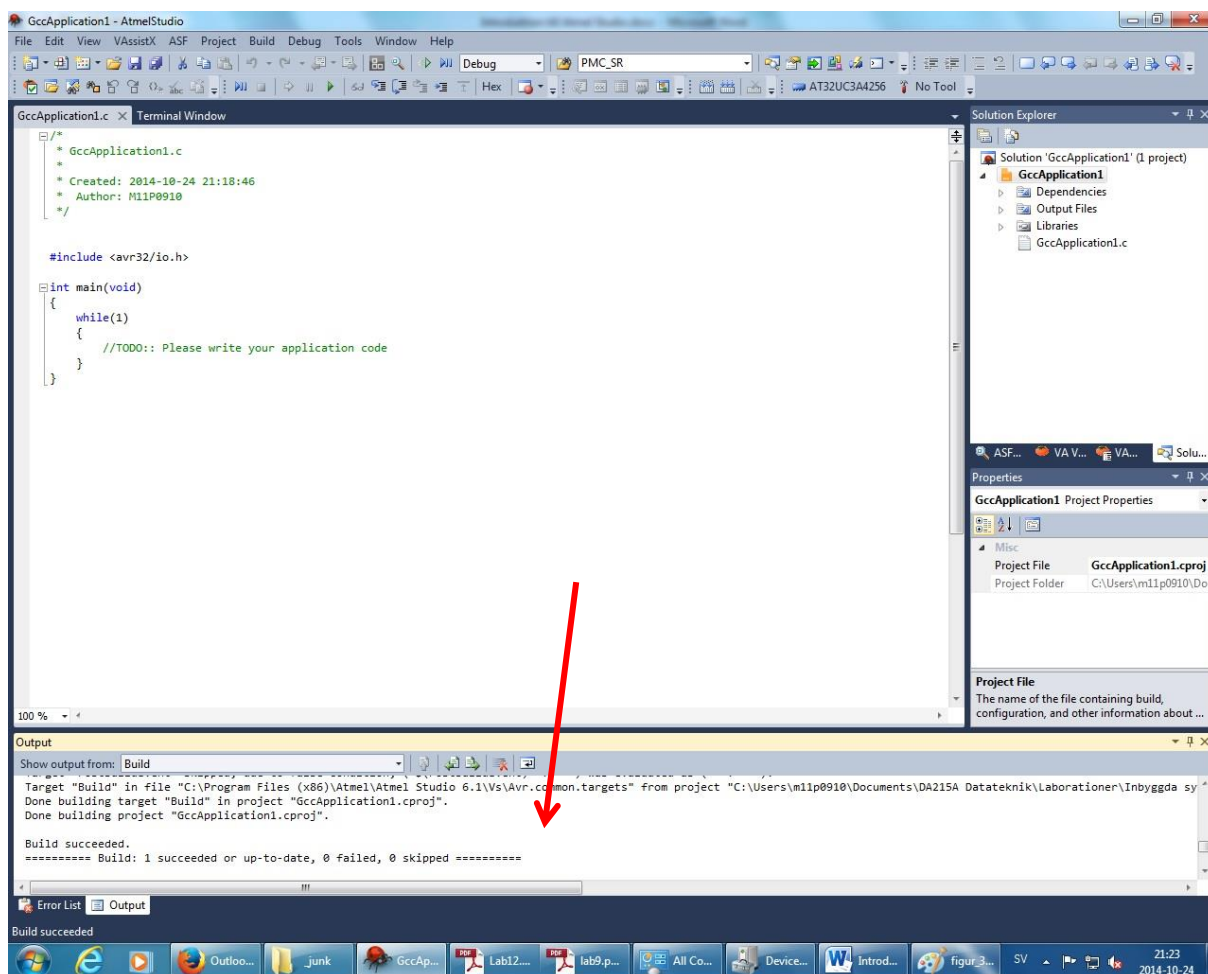


Figur 3-5: Inställningar för C-projekt

# 4 Kompilering och överföring av program

## 4.1 Kompilering

Kompilering av samtliga filer i ett projekt sker när man "bygger projektet". Välj *Build -> Build Solution*, alternativt kan du trycka på F7-knappen. Resultatet av kompileringsprocessen kan man se i fönstret *Output* (se pil i Figur 4-1).



Figur 4-1: Resultat av kompilering

## 4.2 Överföring av program

Innan man kan göra en överföring av ett program så måste man genomföra en konfigurering. Välj *Tools -> External Tools*. Ett nytt fönster öppnas (se Figur 4-2). Klicka på *Add*.

Välj ett namn i rutan *Title*, förslagsvis "Arduino Leonardo". Till höger om rutan för *Command* finns en pil. Där väljer du var filen **avrdude.exe** finns. Leta upp samma mapp där du har installerat Arduinos utvecklingsmiljö och dess drivrutiner. Filen hittar du sedan i mappen **\hardware\tools\avr\bin\**. I nästa ruta, *Arguments*, anger man följande:

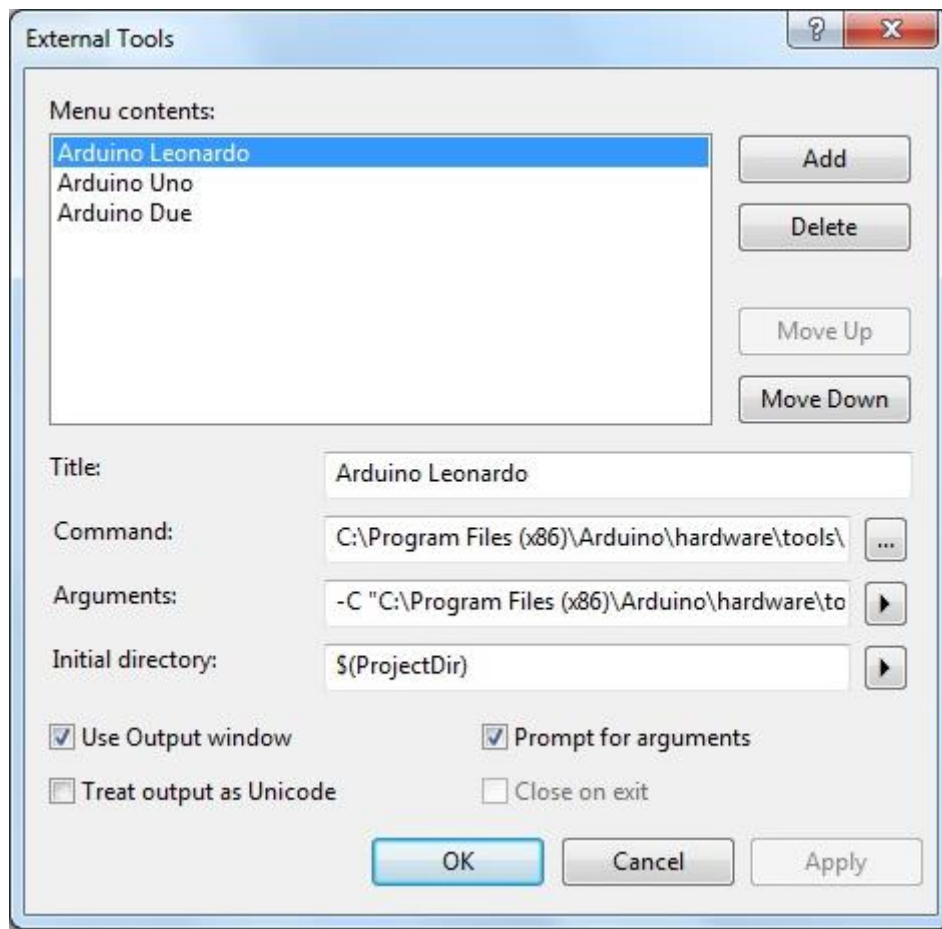
```
-C [sökväg till avrdude.conf] -v -v -p atmega32u4 -c avr109 -P [enhetsbeteckning] -b 57600 -D -U flash:w:"$(ProjectDir) Debug\$(ItemFileName).hex":i
```

Sökvägen till **avrdude.conf** anges genom att ange mappen (där Arduino är installerad) **\hardware\tools\avr\etc\**. Observera att sökvägen måste anges inom citationstecken!  
Enhetsbeteckning är benämningen på COM-porten (ex. COM11). Det är viktigt att denna kommandorad är korrekt! Utgå från textfilen som finns på It's Learning (**avrdude-arguments.txt**).  
Sökväg och enhetsbeteckning måste dock modifieras!

Ange " \$(ProjectDir)" i *Initial Directory*, samt sätt ett kryss i *Use Output window* och *Prompt for arguments*.

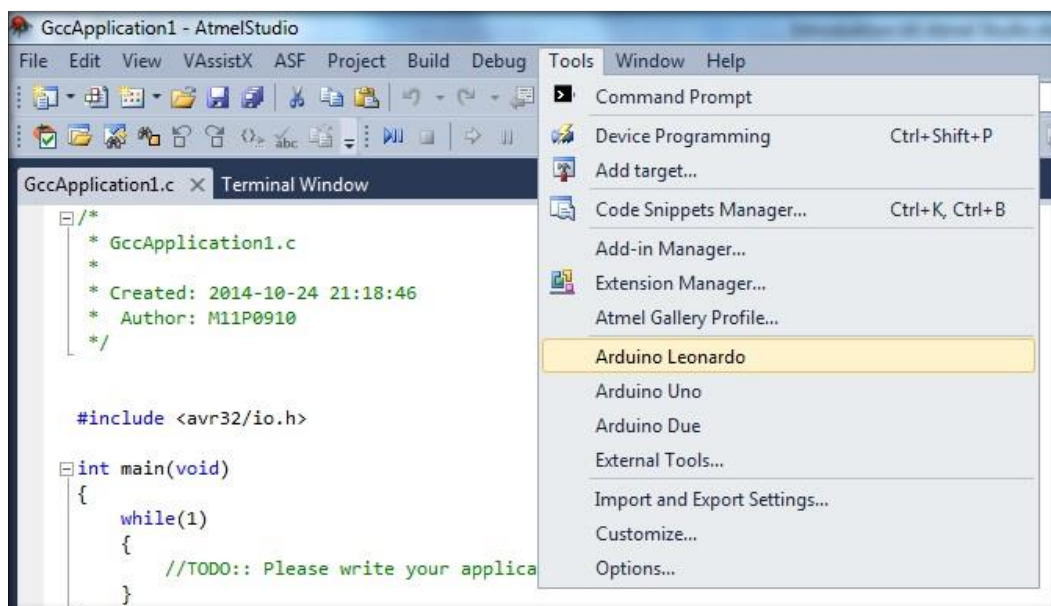
Avsluta genom att klicka på *OK*.





Figur 4-2: Konfigurering av programöverföring

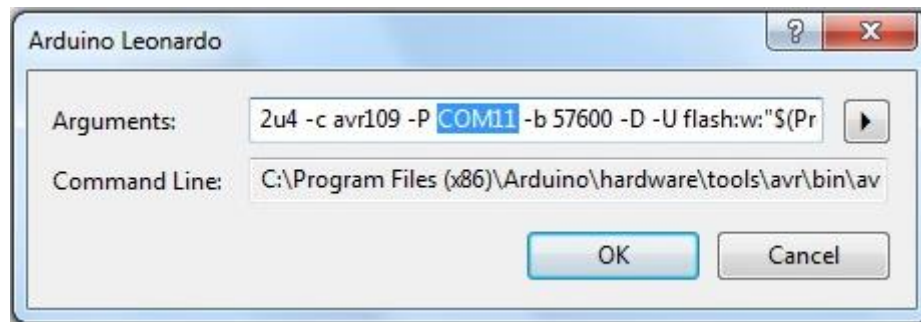
Nu är det dags att föra över programmet till enchipdatorn. Välj *Tools -> Arduino Leonardo* (se Figur 4-3).



Figur 4-3: Välj konfiguration för programöverföring

Ytterligare ett nytt fönster öppnas, tack vare inställningen *Prompt for arguments*. Detta möjliggör att du kan ändra enhetsbeteckning (se Figur 4-4) innan programöverföring, vilket kan vara praktiskt om du kopplar in ett annat Arduino-kort.

Innan du trycker på *OK*, tryck på Arduino-kortets RESET-knapp. Vänta någon enstaka sekund. Nu kan du föra över programmet. När boot-processen är klar kommer ditt program att påbörja sin exekvering på enchipsdatorn.



Figur 4-4: Ändring av kommandorad innan programöverföring