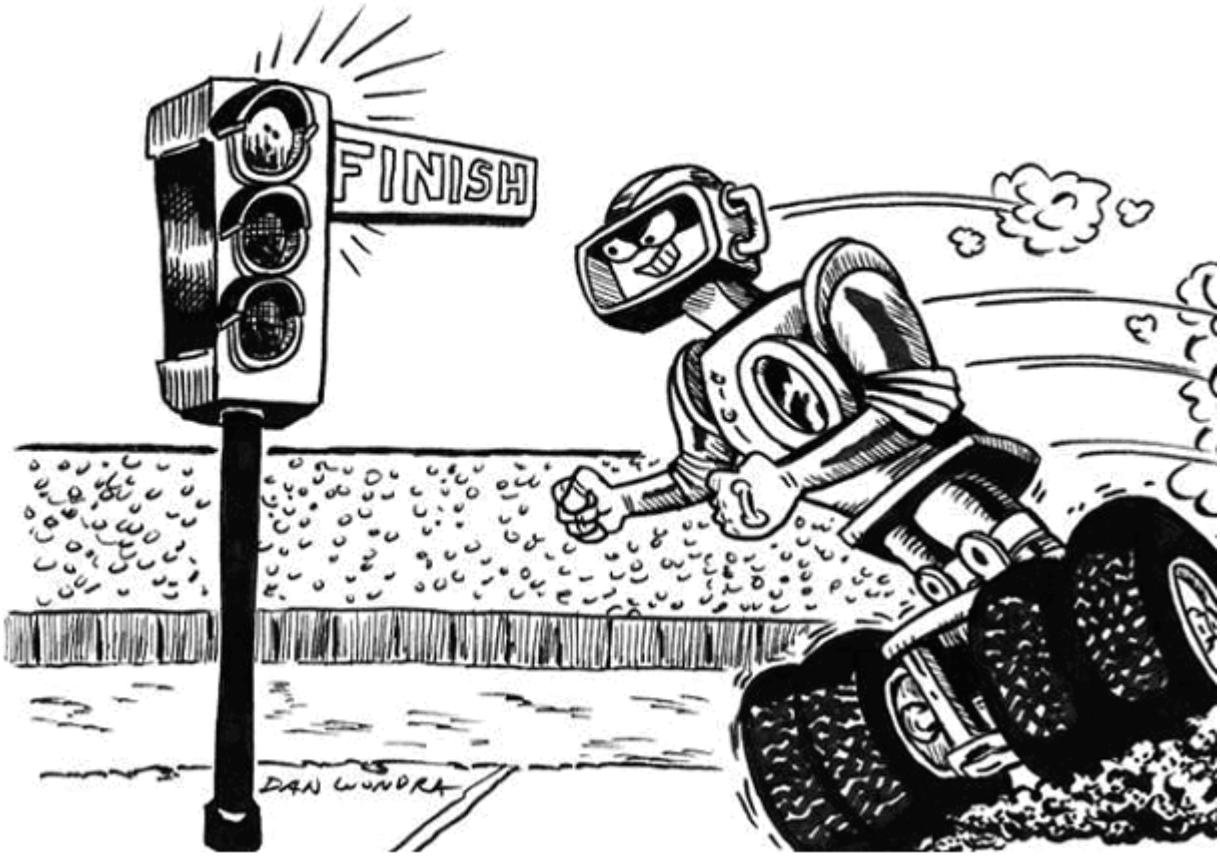
Programming with CEENBoT Commander

Version 1.20, 01-21-2012



CEENBoT

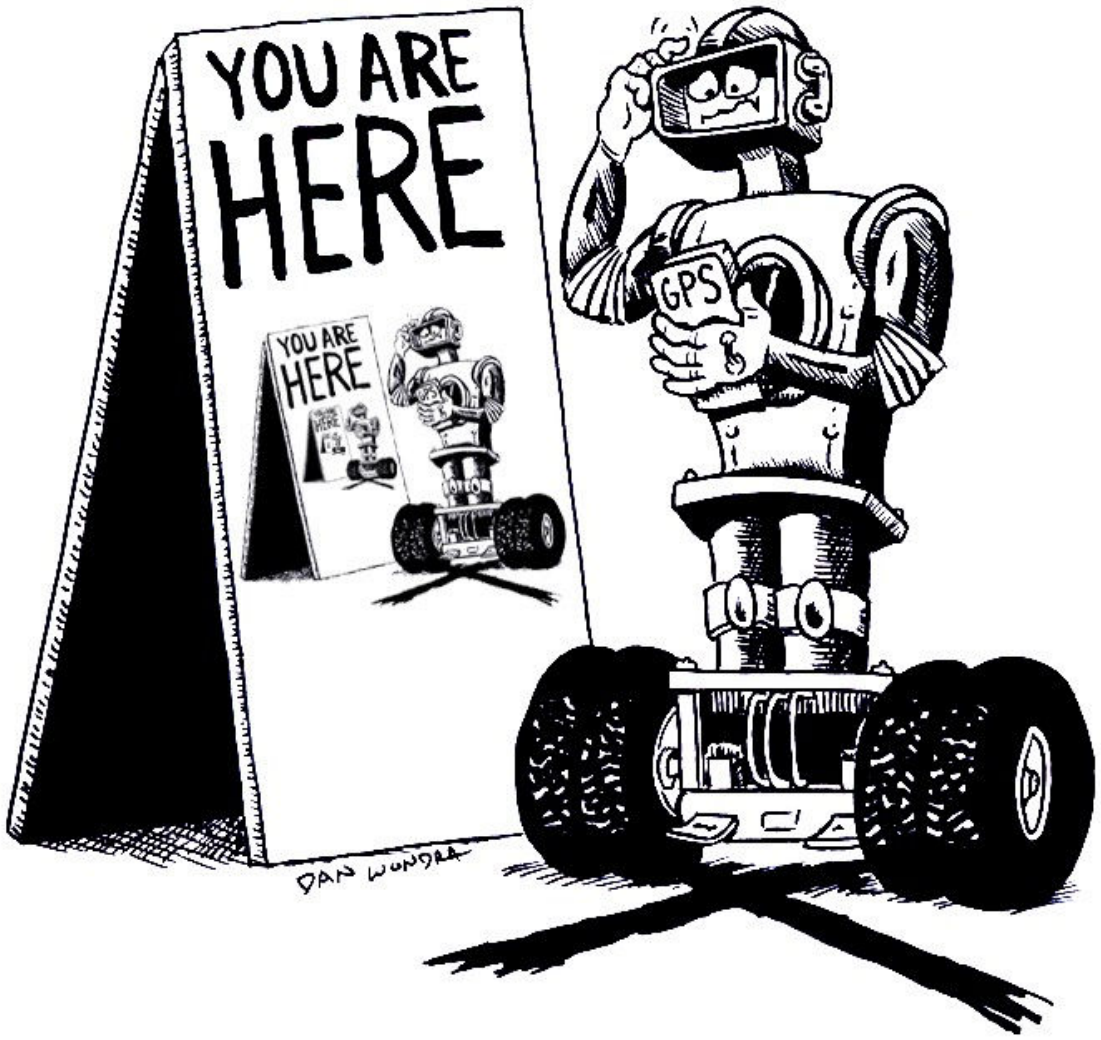
Part 1 – Getting Started

Part 2 – Programming with CEENBoT Commander

Part 3 – Programming Challenges

Table of Contents

Section 1 - Getting Started With CEENBoT Commander



- **Getting Familiar with Your Robot and Programmer**
- **Software Installation**
- **Programmer Installation**

Getting Started

Introduction

This Getting Started guide helps novice robot programmer understand, install and use CEENBoT Commander for basic CEENBoT programming. The guide will familiarize you with concepts, techniques and definitions for programming robotic devices.

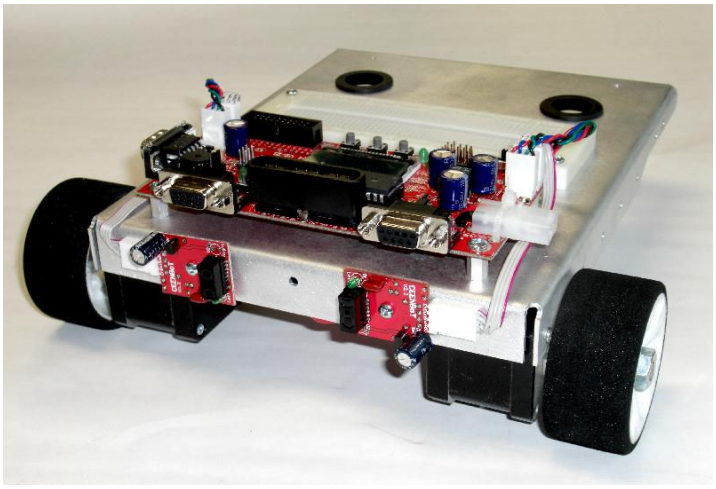
You will be introduced to the hardware programmer device (AVRISP MKii) and CEENBoT Commander graphics program interface (GPI) software which allows you to connect to and program your robot.

What is a CEENBoT?

The CEENBoT is a robot designed and built at the University of Nebraska for use in K-12 classrooms, post secondary education and after school programs.

CEENBoT History

The original purpose of the CEENBoT educational platform was to create interest in the curriculum for engineering students at the University of Nebraska, and improve student retention rates in engineering programs.

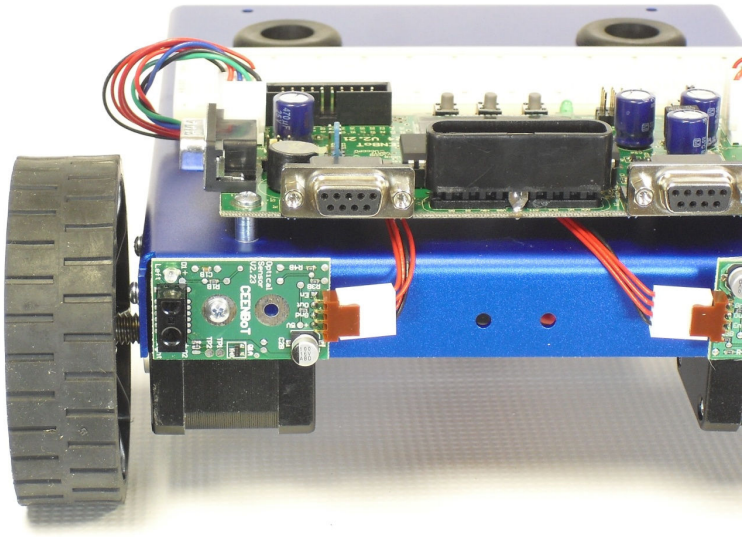


Building upon its initial success at the University, the CEENBoT has been introduced into middle and high schools in Nebraska to interest young learners in science, technology, engineering and math (STEM) and increase the number of students pursuing STEM careers.

The CEENBoT provides students exposure to engineering and programming experiences. Teacher-developed educational materials for grades 5-12 support the use of CEENBoTs for STEM labs and classroom work.

CEENBoT Components:

Although the CEENBoT's design has changed over time, all CEENBoTs have the similar components. The main parts of a CEENBoT are also similar to nearly every other robot you may encounter.



Robot Chassis: The part of the robot where all other pieces are mounted. It provides structure and stability for motors, sensors, electronics, batteries, and optional parts like robotic arms.

Robotic Controller PCB: A printed circuit board containing the microcontrollers and the electronics to control motors, display information, and interact with its surroundings.

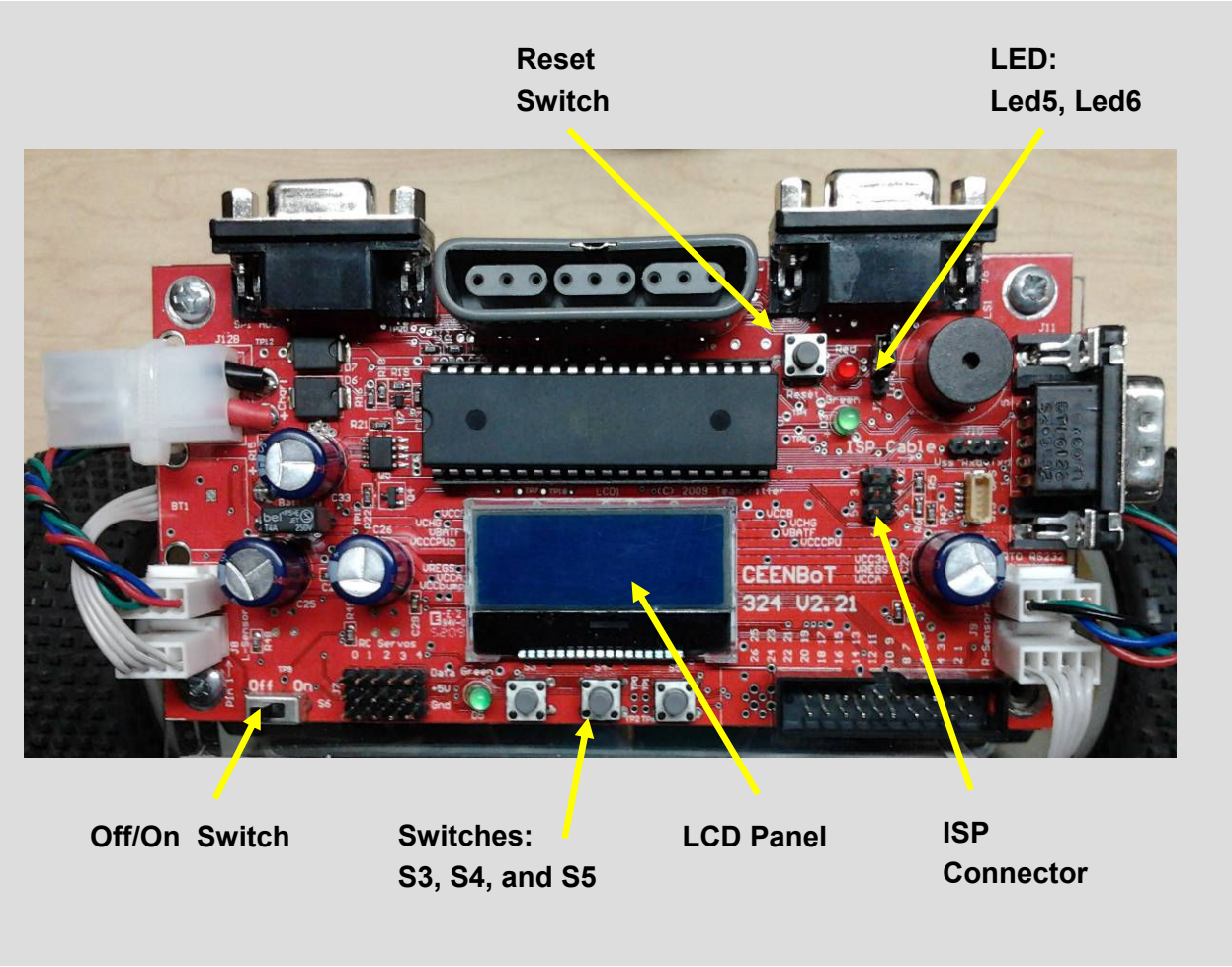
Infrared Proximity Sensor: A device that emits an infrared beam and detects the reflection of from objects in front of the robot. A common use is to avoid collisions with objects or walls or to navigate a maze.

Motor: A motor turns electricity into rotation of a shaft. Put wheels on the shaft and your robot can move under your control. CEENBoTs use stepper motors, which turn in small, consistent angular movements rather than continuously. The CEENBoT stepper motors turn 1.8° with each step. A full revolution takes 200 steps.

Wheels: Plastic wheels with rubber tread turn the rotation of the stepper motors into motion.

CEENBoT Robotic Controller PC Board

The microcontroller board on the CEENBoT is based on the Atmel ATmega324P microprocessor. The board has switches, lights connectors and a display to give you several ways to interact and work with your robot.



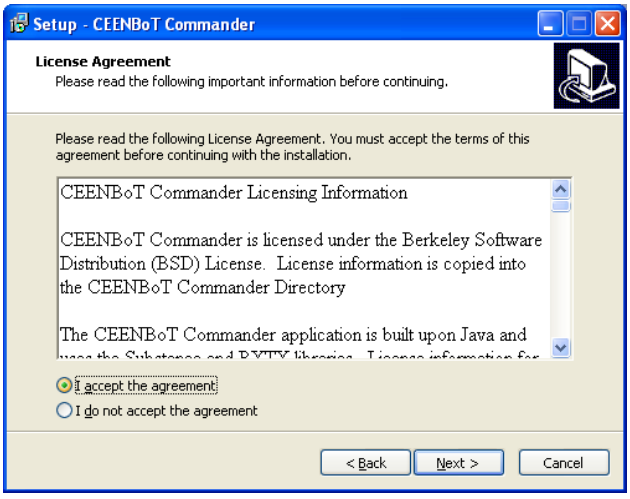
Qty	Feature	Purpose	Programmability
1	Off/On switch	Turn robot on and off	
1	LCD panel	Show text and graphics	Display from program
3	Input switches: S3, S4, S5	User input from switches	Read by program
1	ISP connector	Align pin 1 with red stripe	
2	LEDs: D2, D3	Programmable	Control from program
1	Reset switch	Restart controller software	

Installing the CEENBoT Commander Program for Windows

Download and install the setup program to your hard drive.

Double click the downloaded file and follow the prompts. Click next.

Accept the license agreement. Click next.

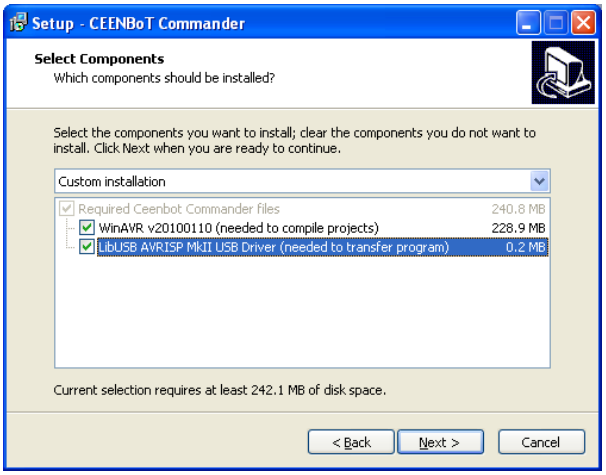


Check both CEENBoT Commander components **UNLESS** you have AVR Studio already installed on your computer.

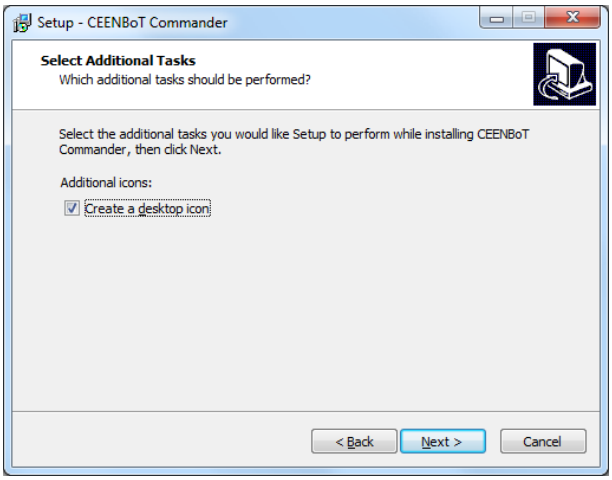
1. WinAVR v20100110
2. AVRISP MkII USB Driver

Note: The CEENBoT Commander AVRISP MkII USB Driver is incompatible with the driver provided with AVR Studio. Contact us if you have this situation.

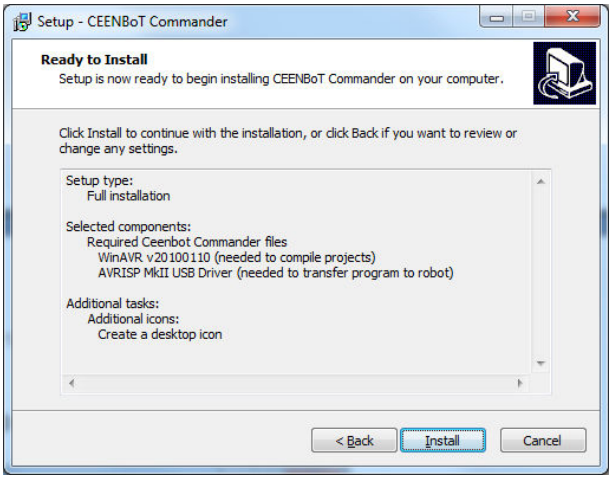
Click “Next”.

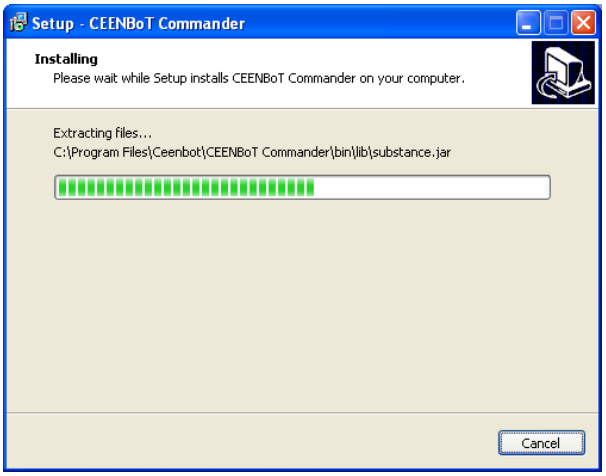


Allow the application to create a desktop icon. Click next.

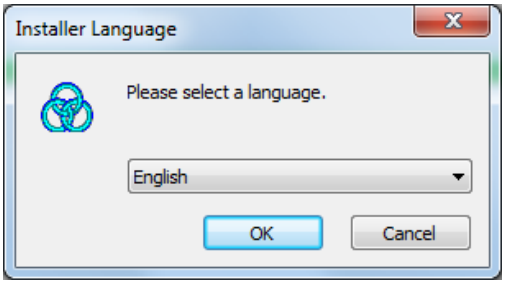


The setup process will display a Ready to Install box which explains what applications and tasks will be installed on your computer. Click next.





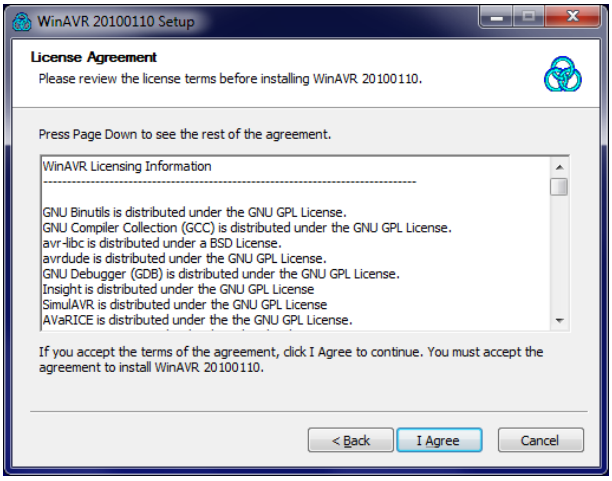
The setup process will ask you to select the language. Click OK.



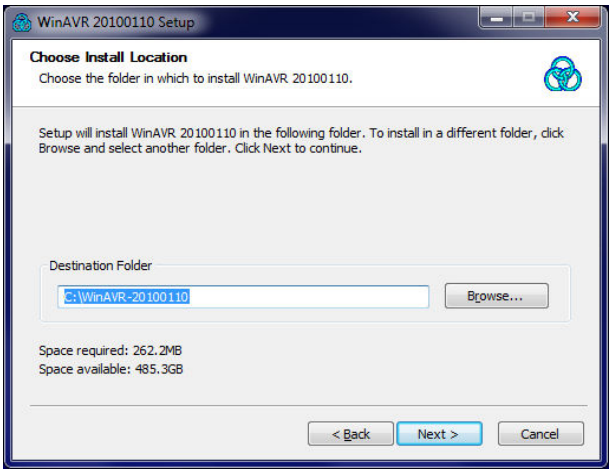
The Commander installation should be complete.

Next you will be prompted for the WinAVR 20100110 installation. Click next.

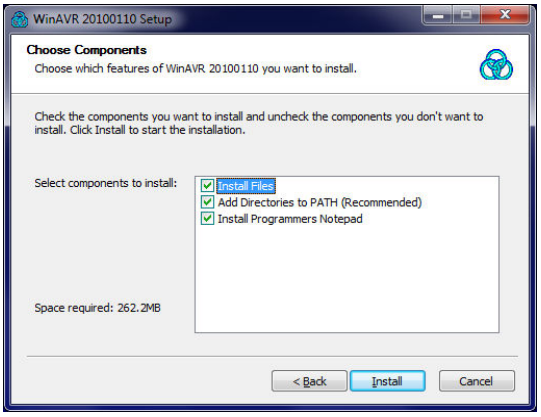
Agree to the software license agreement.



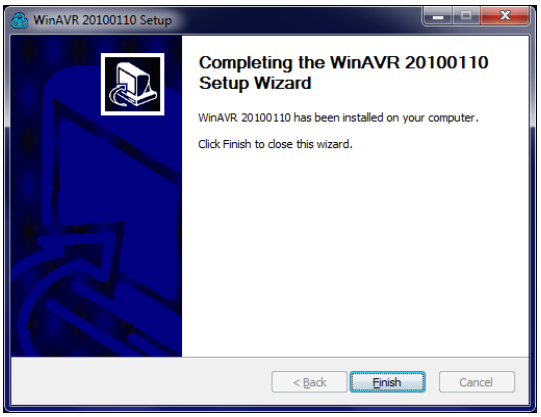
Use the default destination folder to install WinAVR 20100110.



Choose the components of WinAVR 20100110 you wish to install. We recommended you select all the components for installation.



The setup program will install WinAVR 20100110. Installation is complete.



Click finish to close the Commander setup application.

After installation completes, a web page will open to display information about WinAVR. Most people will not need this information and the web page may be closed.

You have now installed the Commander, WinAVR, and AVRISP MKII applications and are ready to start programming your CEENBoT.

Installing the CEENBoT Commander Program for Apple OSX Snow Leopard

CEENBoT Commander is supported on Apple OS X version 10.6 (Snow Leopard) on all machines and version 10.5 (Leopard) on 64 bit machines. The operating system and Java should be updated with current patches.

CEENBoT Commander is delivered in a .dmg installation file. Contained in that file you will find two components:

1)The open source package CrossPack AVR development tools, and

2) The CEENBoT Commander program

Download the CEENBoT Commander Install .dmg file to your hard drive.

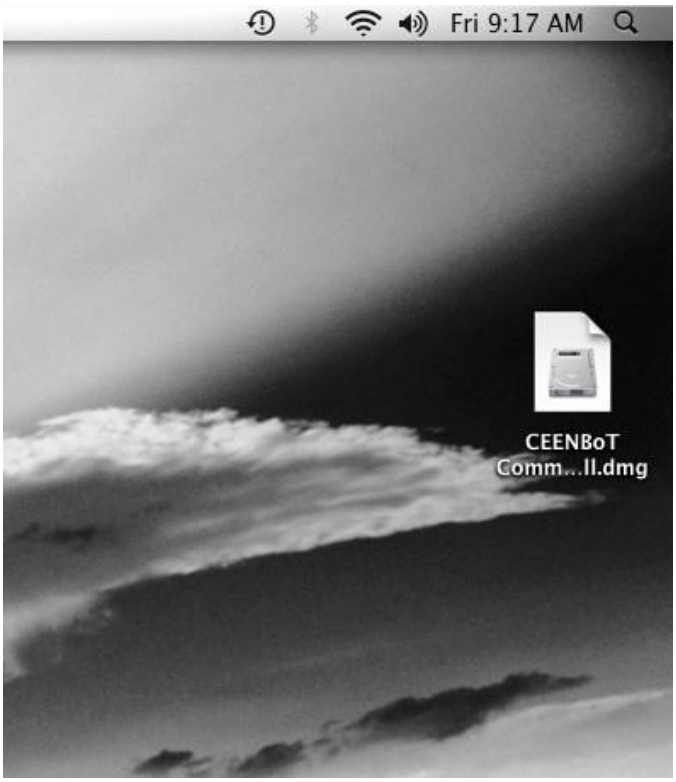


Figure 1 - CEENBoT Commander dmg file on desktop

Double-click on the CEENBoT Commander Install .dmg file to open it on your computer. The newly opened file will appear as a folder on your desktop.

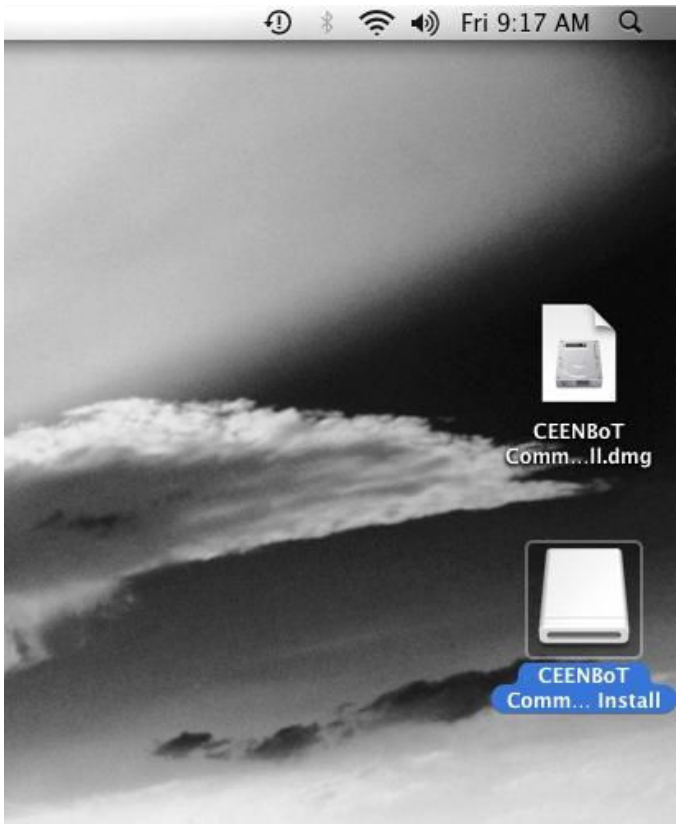


Figure 2 - CEENBoT Commander Install Folder

Open the CEENBoT Commander Install folder.

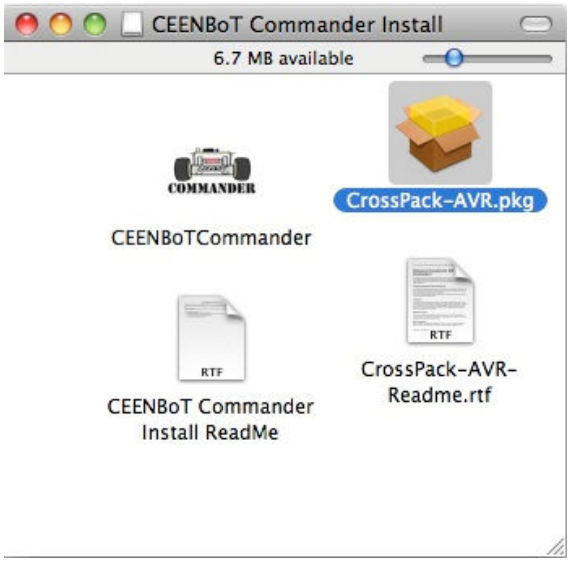


Figure 3 - CEENBoT Commander Install Folder Contents

Double-click the CrossPack AVR by double-clicking on CrossPack-AVR.pkg.

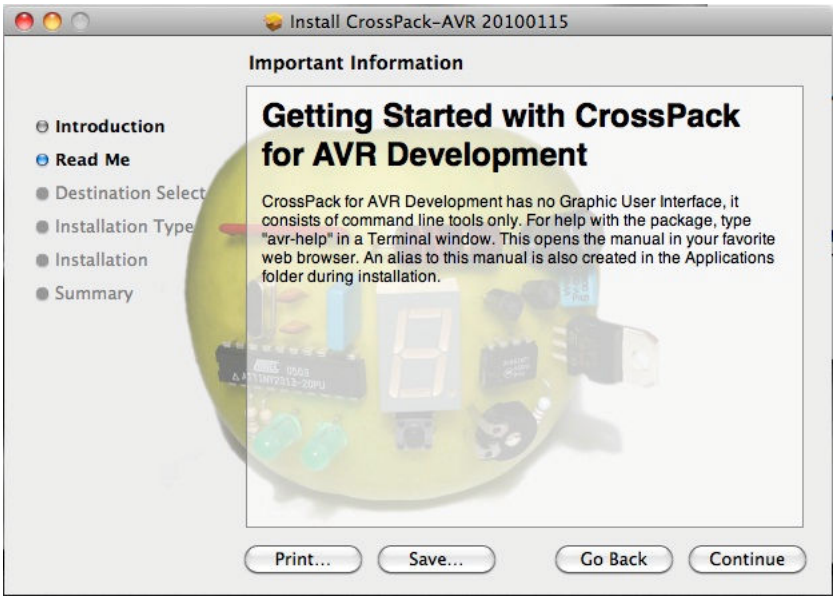


Figure 4 - CrossPack Installation Initial Screen

Install CrossPack in the default location

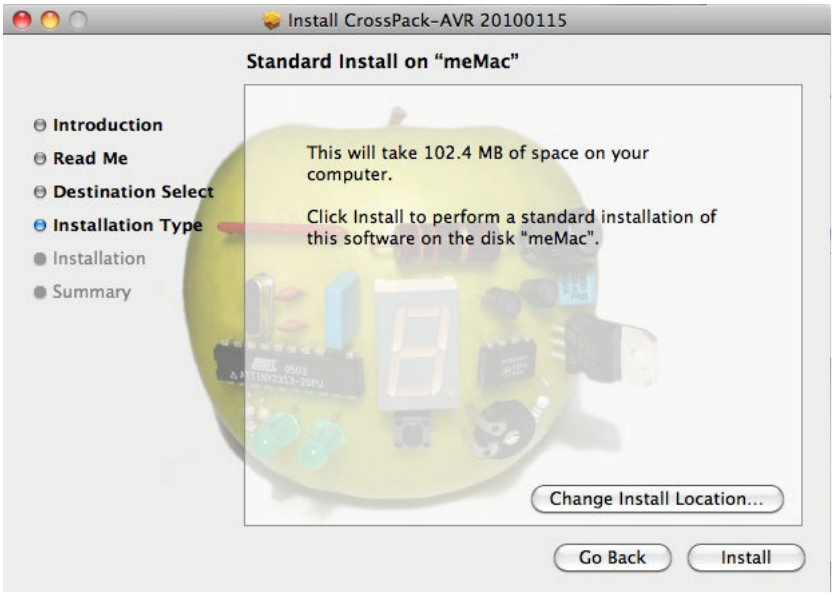


Figure 5 - CrossPack Configuration / Install Screen

The CrossPack installation program will show its progress.

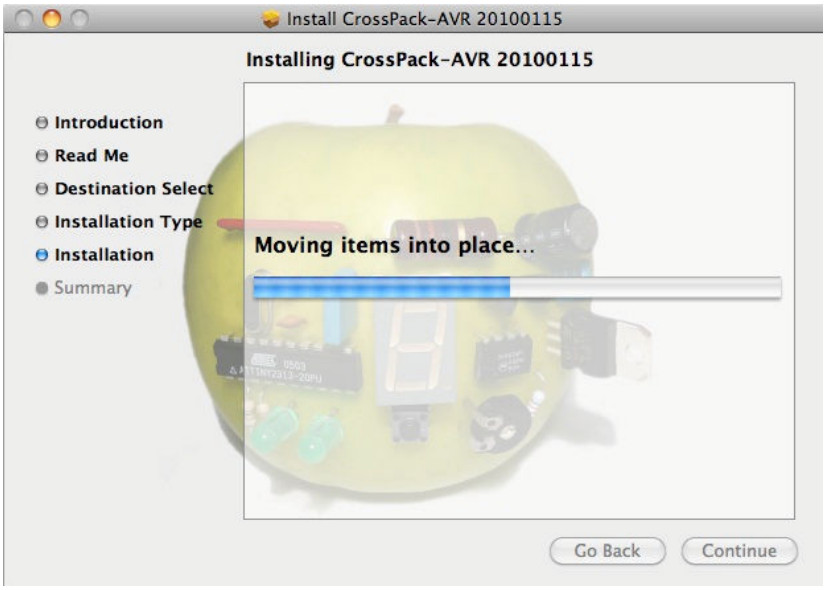


Figure 6 - Crosspack Progress Screen

The CrossPack installer will show a success screen when done.

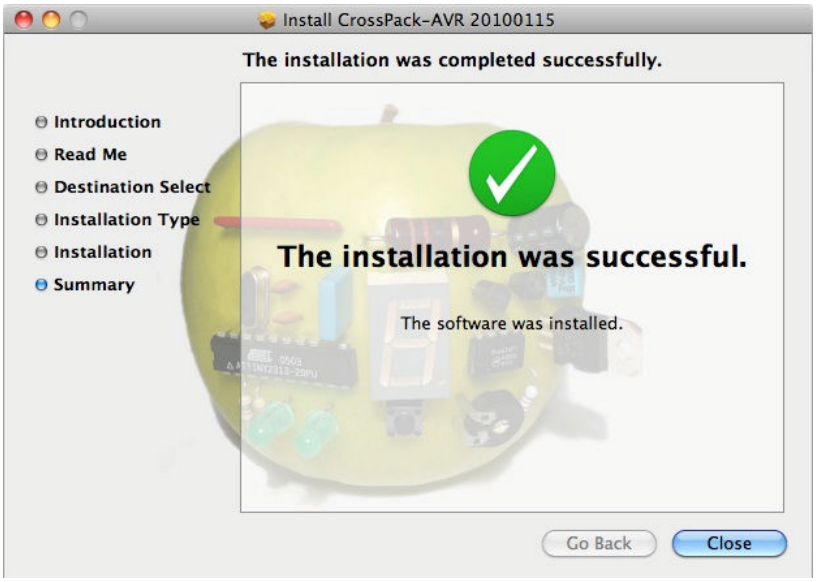


Figure 7 - Crosspack Completion Screen

Drag the **CEENBoTCommander** icon to your Applications folder.

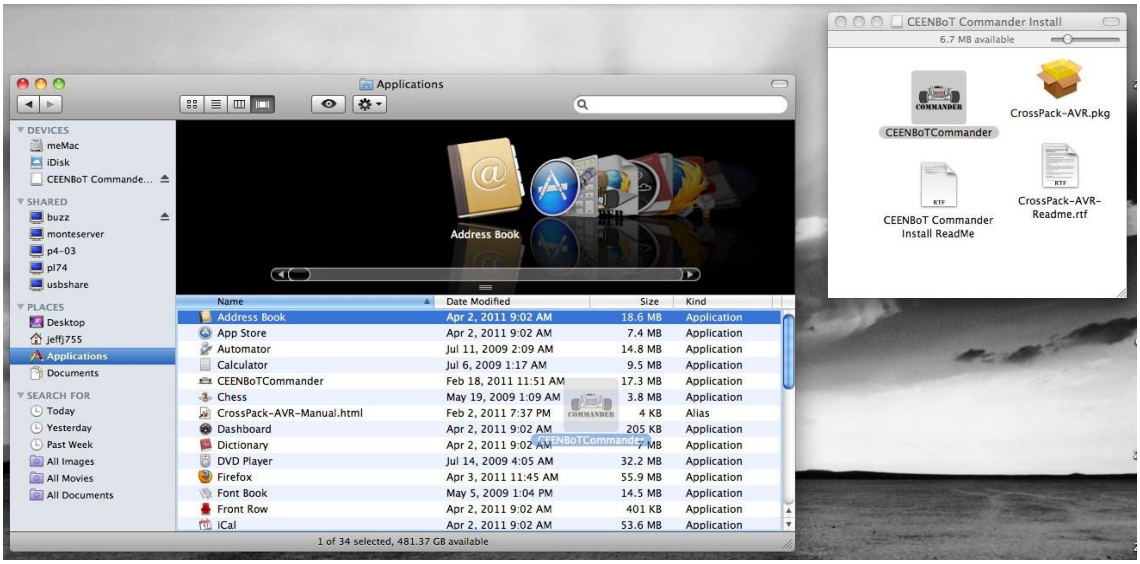


Figure 8 - CEENBoT Commander Drag and Drop Installation

You have now installed Commander for Apple OSX.

To run CEENBoT Commander, double-click on **CEENBoTCommander** in the Applications folder.

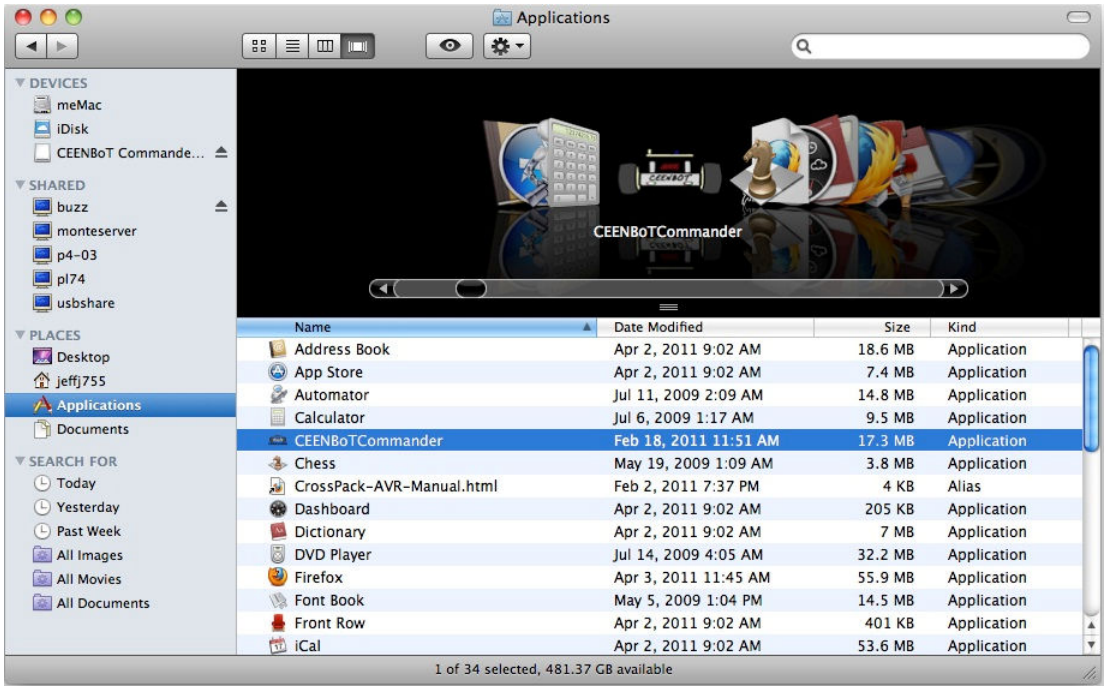
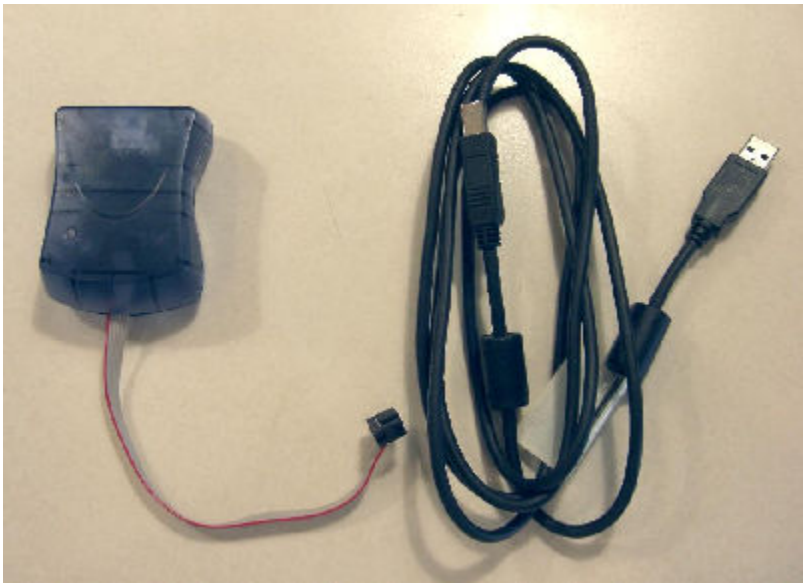


Figure 9 - Running CEENBoT Commander

AVRISP Programmer



The AVRISP Programmer allows the computer to send new programs to the CEENBoT. Each time you send a new program to the robot, the old program is overwritten. The AVR Programmer connects between a computer USB port and the CEENBoT in-system programmer (ISP) port.

The AVRISP programmer requires a software driver to be installed on your computer.

On Windows machines, the programmer requires the driver to be set up for each unique AVRISP mkII programmer (for example in a classroom) and USB port combination. We recommend using the programmer in the same USB port each time.

The software driver is installed at the same time as CEENBoT Commander. If you have not yet installed CEENBoT Commander, please install it now. Your computer will not recognize the programmer when it is plugged in without first completing the installation.

Setting up the AVRISP programmer driver:

1. Connect the USB cable into the USB connector on your computer.

2. Plug the other end of the USB cable to the AVRISP programmer. Once connected for the first time, the computer will recognize the new device and install the software driver. This may take a couple minutes to complete.

3. Connect the ISP cable to the ISP connector on the PC board. Make sure pin 1 on the ISP plug (red stripe) is lined up with pin 1 on the PC board (see figure 4).

The AVRISP programmer is ready to use when the LED near the robot is lighted **green** and no other colors or LEDs are lighted

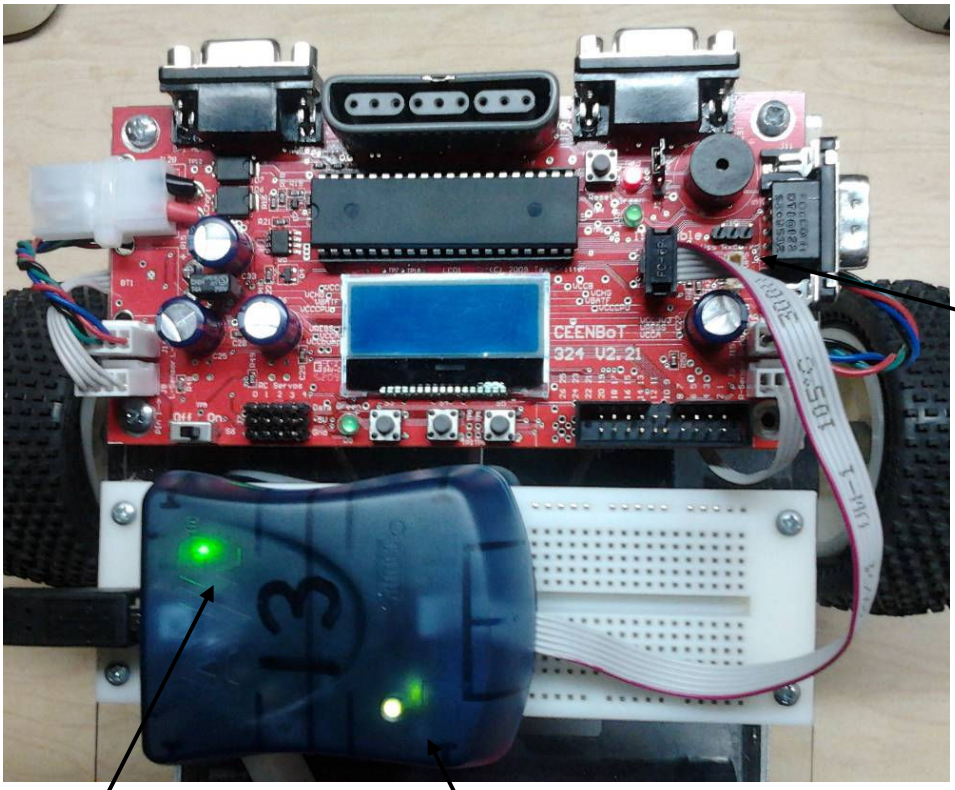
Computer USB Status Light

No Light	Normal inactive state. The LED may flash when the programmer is first plugged in and then go out.
Flashing Green	Sending data.

Robot ISP Connection Status Light

No Light	Programmer has not been connected to a robot before, or drivers not installed correctly.
Green	Connected to the robot, and the robot is turned on.
Orange	Programming the robot. Do NOT disconnect any cables while orange.
Red	Connected to the robot, but the robot is turned off.
Flashing Orange	Pin 1 (red stripe on cable) not aligned with pin 1 on PC Board.

CEENBoT ISP connector and AVR Programmer



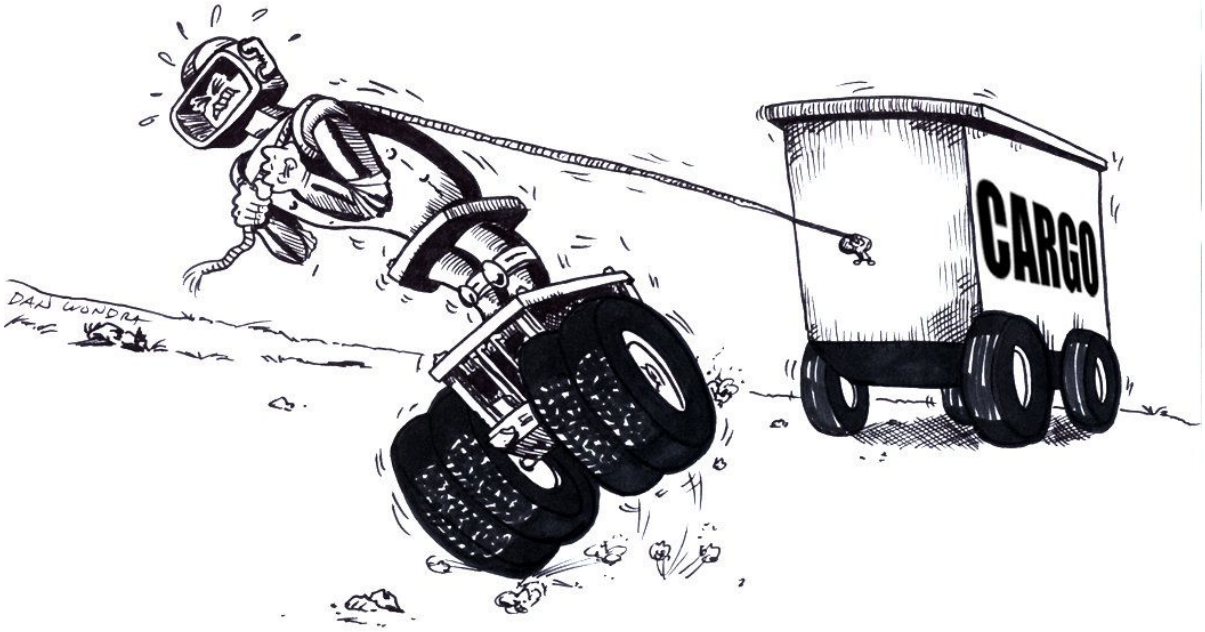
ISP Connector

Align pin 1 with Red Stripe on cable

Computer USB Status

Robot ISP Connection Status

Section 2 - Programming With CEENBoT Commander

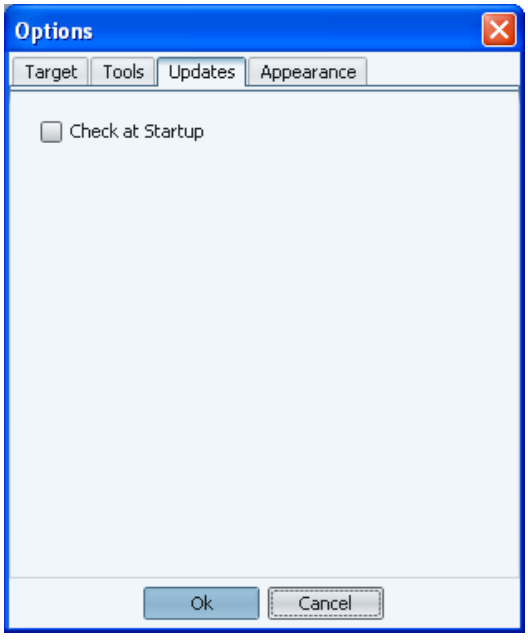


- **Getting Familiar with CEENBoT Commander**
- **Starting to Use CEENBoT Commander**
- **Programming Tools**
- **Programming Techniques**
- **Configuring CEENBoT Commander Options**

CEENBoT Commander Graphical Program Interface (GPI)

This Programming Guide will show you the parts of CEENBoT Commander you will want to know about, the commands that can be used to create your programs, and the settings for each command to make your robot behave the way you want. CEENBoT Commander is a graphical program interface, which means you can write a program by dragging and dropping commands onto a window rather that writing programs with words. Start CEENBoT Commander by clicking or double-clicking on the program icon.

As the program is loading, a splash screen will display. On the first startup after installation, you may be prompted whether you wish to always check for program updates when Commander starts.

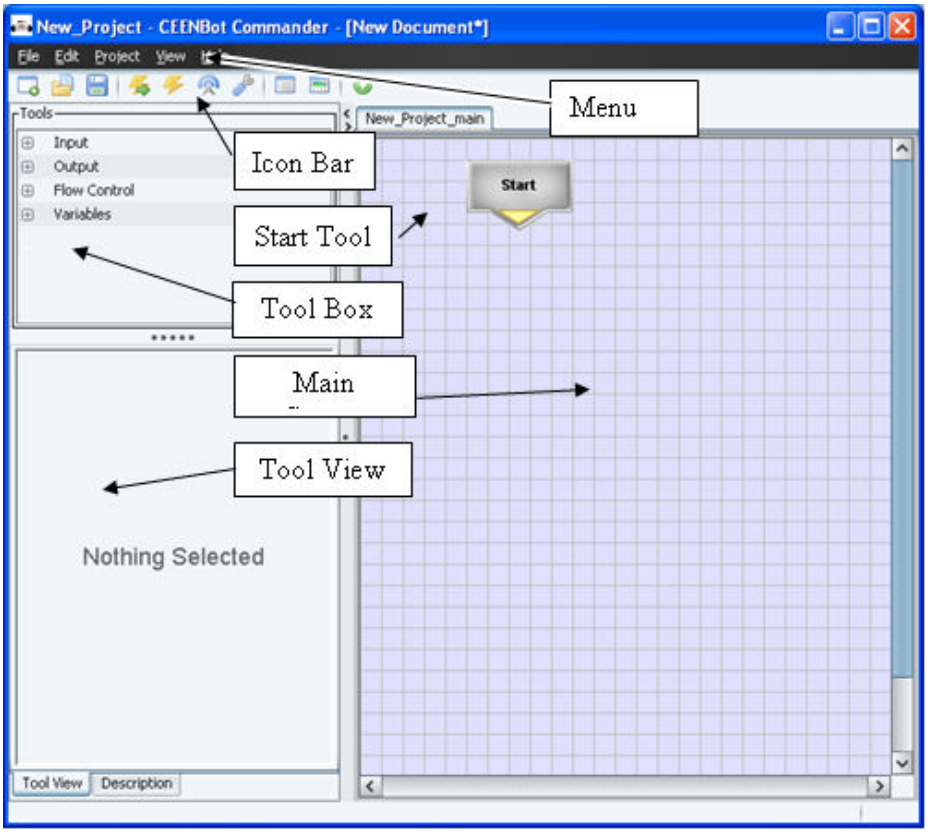


Subsequent startups will not ask this question although you can change your preference in the "File" > "Options" > "Updates" window.

GPI Overview

The CEENBoT Commander screen contains several areas that are used when programming

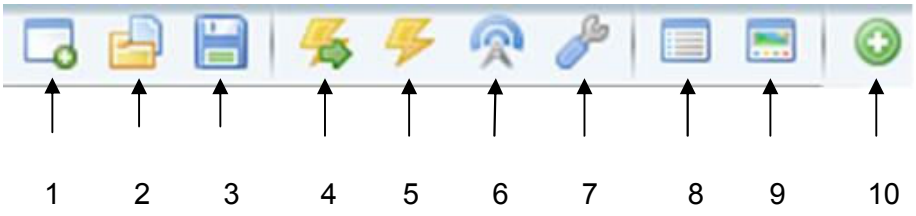
- Menu Bar Menu based controls grouped by File, Edit, Project, View, and Help.
- Icon Bar Buttons that provide one-click access to frequently used controls.
- Tool Box Programming commands are grouped by Input, Output, Flow Control, and Variables.
- Main Window where the tool program commands are arranged into programs.
- Tool Window
    - Tool View Show and change the arguments of the selected tool program element.
    - Description Helpful information on how to use the selected tool.



Using CEENBoT Commander

CEENBoT Commander is designed to simply and easily program a CEENBoT robot. Most activity starts at the Icon Bar, where you can create a new project, open an existing project, save your work, and program your robot.

Icon Bar Features

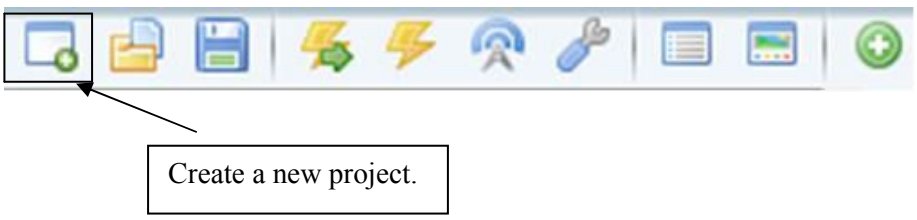


1. **Create** a new project
2. **Open** an existing project
3. **Save** the current project
4. **Build and upload** project to robot
5. **Build** the current project
6. **Upload** project to robot
7. **Options**
8. **View C code**
9. **View graphically**
10. **Add a new module** to the current project

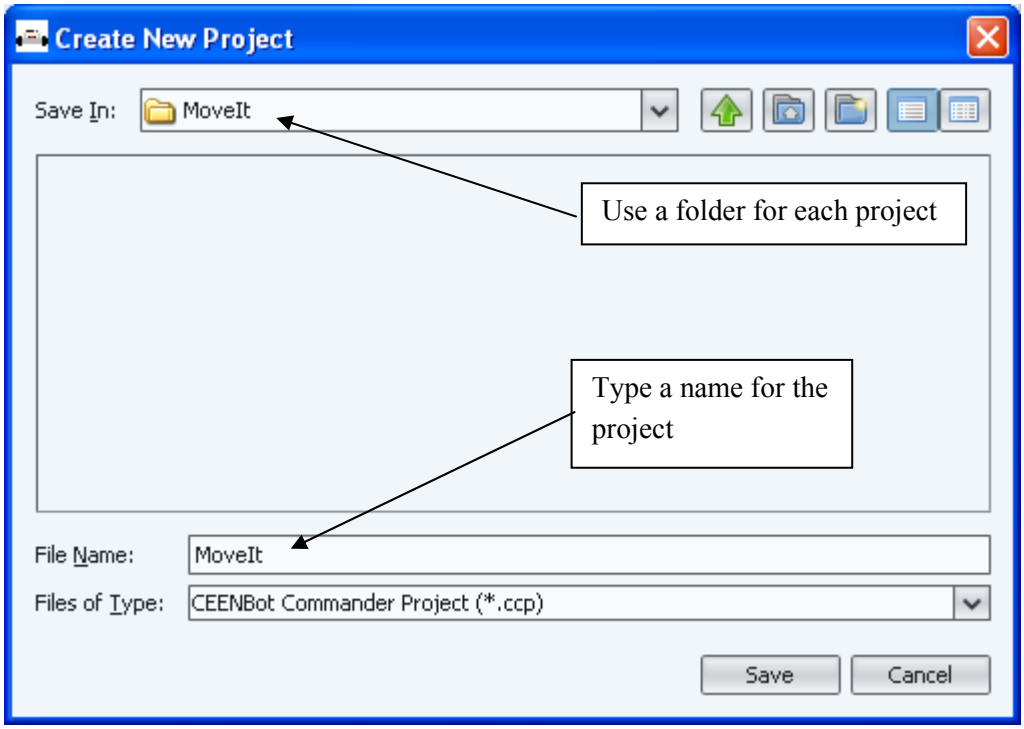
In the next few steps, we show you how to start writing your program and send it to your CEENBoT robot. We begin with creating a project. A project contains the main program and any modules that make up the entire set of instructions for a robot.

Creating a New Project

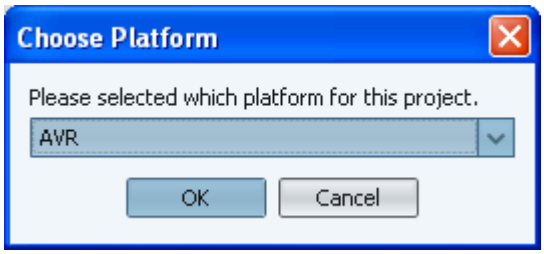
Click on the icon bar:



Select or create a folder to hold the project. Type a file name for the project. The name can be the same as the project folder, or could be something different.



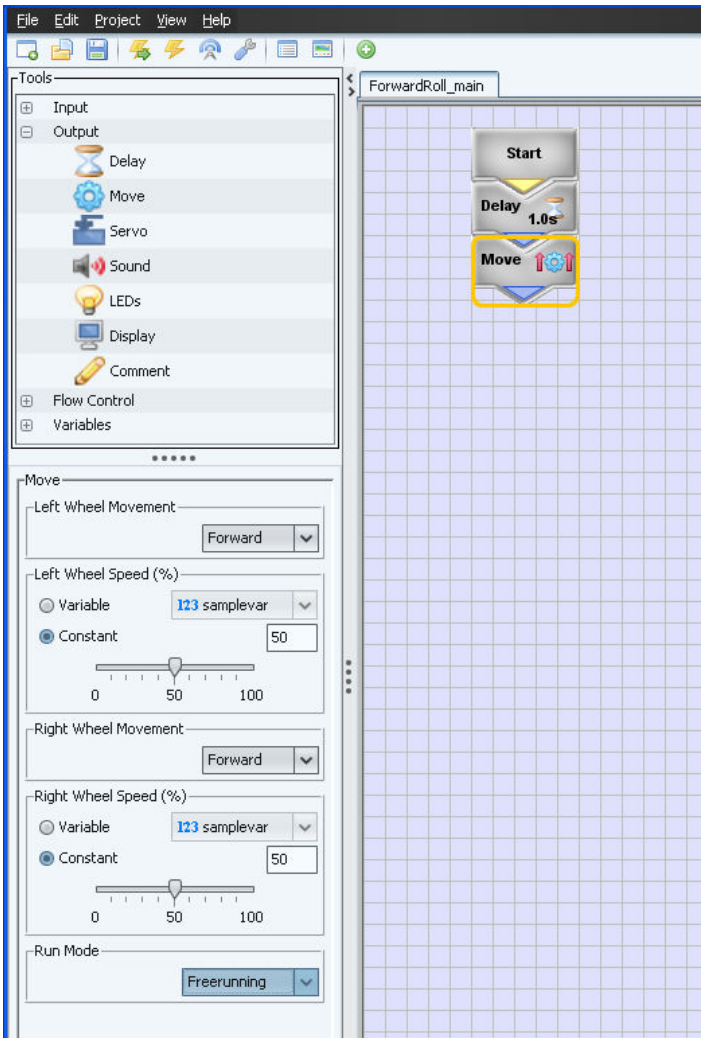
Choose AVR as the platform. CEENBoT Commander is designed to work with different robots. Your robot is based on the AVR family of microcontrollers. This should not be changed.



Create a Program

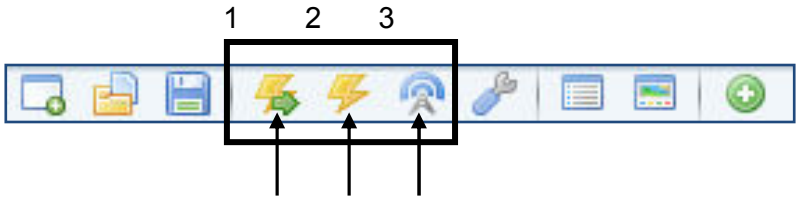
> **NOTE: Programs Start Running Immediately!**
>
> - Programs uploaded to the robot start immediately after they are loaded.
> - The robot wheels should not be touching the floor or table top while programming.
> - Place the robot on a block of wood or have your partner hold the robot. Otherwise the robot may start moving unexpectedly.

- Open the **Output** group in Tools.
- Drag and drop the **Delay** tool and attach it to **Start**.
- Click on the **Delay** tool block and enter 1000 milliseconds in **Tool View**.
- Drag and drop the **Move** tool and attach it to **Delay**.
- Click on the **Move** tool block and enter move arguments in **Tool View** as shown.



Build and Upload Project to Robot

Now that you've created your program, you will want to save and transfer it to your robot. The Build and Upload icons will allow you to do this.



1. Build and upload project to robot
2. Build the current project
3. Upload project to robot

Pre-Upload Checklist

- CEENBoT Commander will ask you to name your file and project if you have not already named it.
- CEENBoT Commander will save your file automatically before building it.
- Be sure the robot wheels are not touching the floor or table top
- Be sure the robot is powered on
- Be sure the AVR Programmer is properly connected to the computer and robot
- Be sure the robot status light on the AVR Programmer is green

When everything is ready, click on the icon bar: Build and upload project to robot. The computer USB status light will start blinking and the robot ISP status light will turn orange.

After the program is transferred, the robot will start running it. Watch the robot while testing your program. Remove the ISP connector so the robot can run freely.

The program will run one time after the upload finishes. To run the program again, press the Reset button on the PC board or turn the robot off and on.

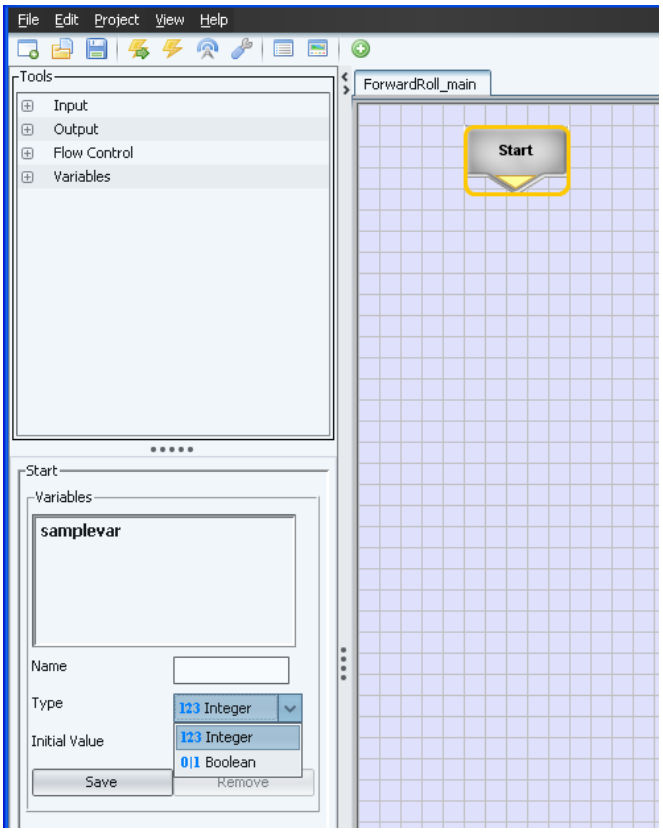
NOTE: Programs are Stored in Flash Memory

The robot can be powered off since the robot programs are saved in the robot flash memory. When the robot is powered on, this program will start running immediately.

CEENBoT Commander Tools

Start Tool

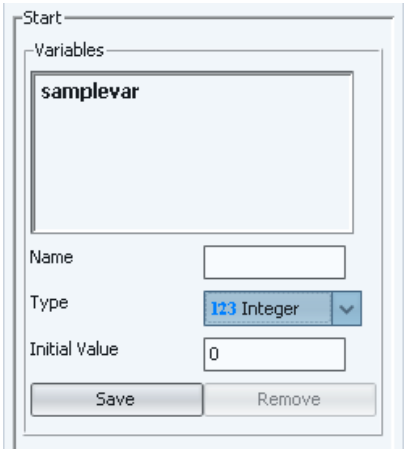
The Start Tool marks the beginning of the program chain. All other tools attach to it. CEENBoT Commander automatically places the Start tool on a new project.



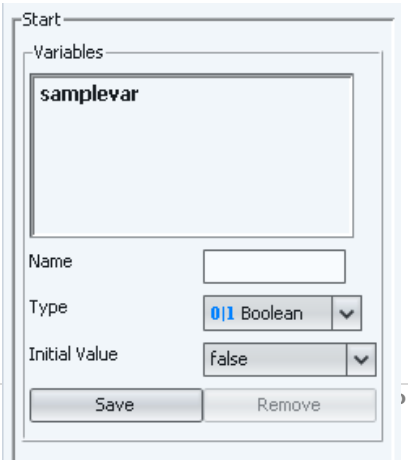
Variables are also added to the program using the start tool. These variables can be used by other tools.

There are two kinds of variables: Integer and Boolean.

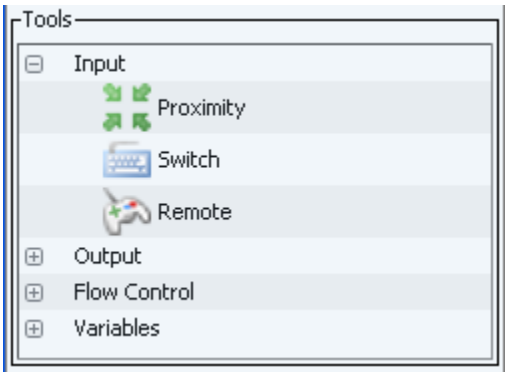
Integer values must be in the range -32768 up to 32767 with no decimal..



Boolean values are either TRUE or FALSE, and the two values are interpreted as False = 0 and True = 1.

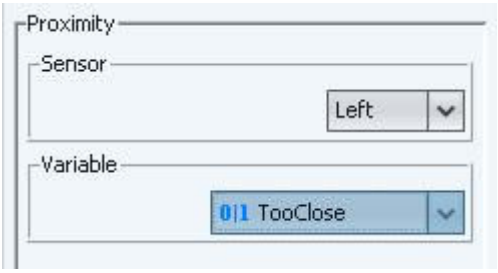


Input Tools



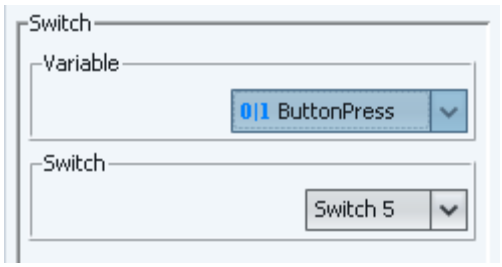
Proximity Tool:

Reads the status of left and right infrared proximity sensors and copies this into a Boolean variable. If the sensor is obstructed, the variable will contain a value of 1 (e.g. true). If the sensor is unobstructed the variable will contain zero (e.g. false).



Switch Tool:

Reads the status of one of three input switches on the controller board, and copies this into a Boolean variable. If the switch is depressed, the variable will contain a 1 (e.g. true). If the switch is not depressed, the variable will contain zero (e.g. false). The switch number corresponds to a switch on the physical board. See chart below.

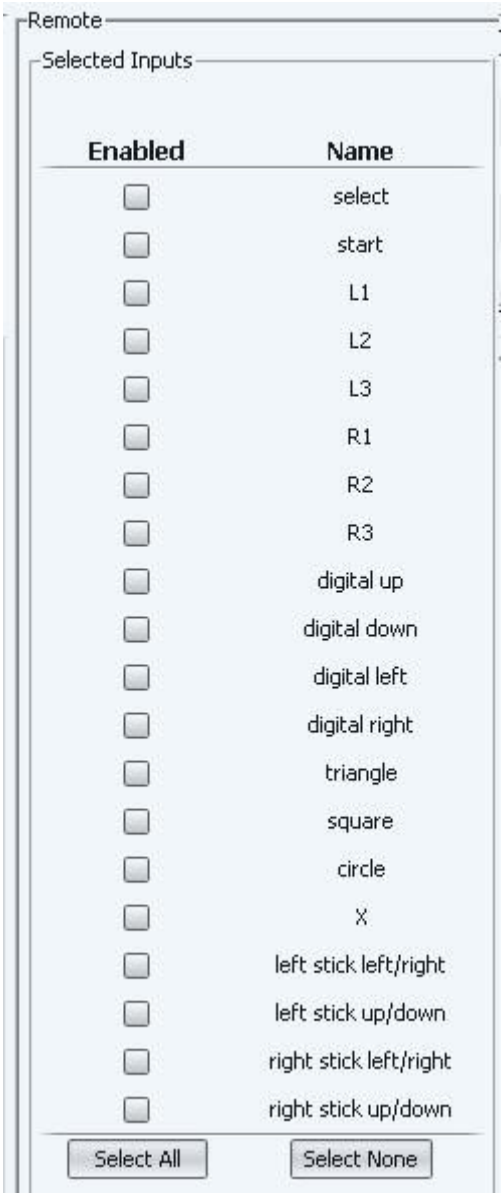


Platform	Switch	Label	Position
'324	3	S3	Left
	4	S4	Center
	5	S5	Right

Remote Tool:

Reads the values received from the Playstation remote control and puts selected values into pre-defined variable (pre-defined means you don't have to set up these variables in the Start tool.

To use the remote tool, drop it onto the main screen. Check the remote input or inputs you want to use, and the variables will appear in variable selection boxes along with the other variables you created.

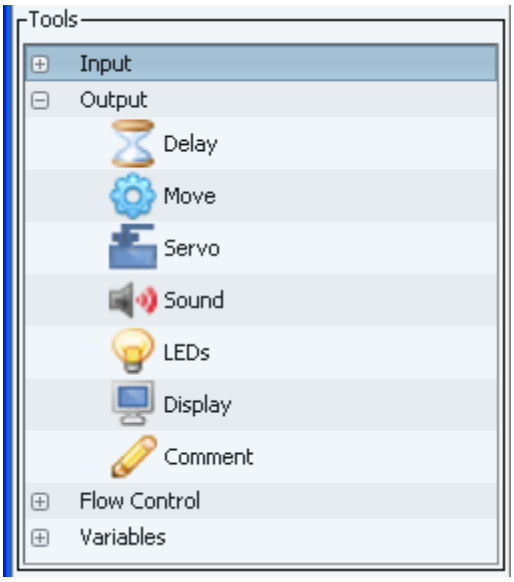


Some remote variables are Boolean (true or false) and others are integers

Boolean Variables (True or False, 1 or 0)		**Integer Variables** (-128 … 127)	
Button	***Variable Name***	***Button***	***Variable Name***
select	PSX_select	left stick left / right	PSX_lstick_lr
start	PSX_start	left stick up / down	PSX_lstick_ud
L1	PSX_L1	right stick left / right	PSX_rstick_lr
L2	PSX_L2	right stick up / down	PSX_rstick_ud
L3	PSX_L3		
R1	PSX_R1		
R2	PSX_R2		
R3	PSX_R3		
digital up	PSX_dup		
digital down	PSX_ddown		
digital left	PSX_dleft		
digital right	PSX_dright		
triangle	PSX_triangle		
square	PSX_square		
circle	PSX_circle		
X	PSX_X		

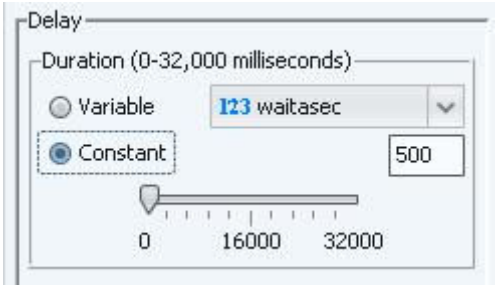
To use the left and right stick controls with integer variables and analog outputs (from -128 to 127), **the remote MUST be in Analog mode** and the red light below the “Analog” button on the remote must be illuminated.

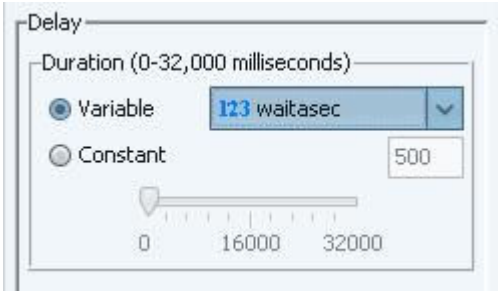
Output Tools



Delay:

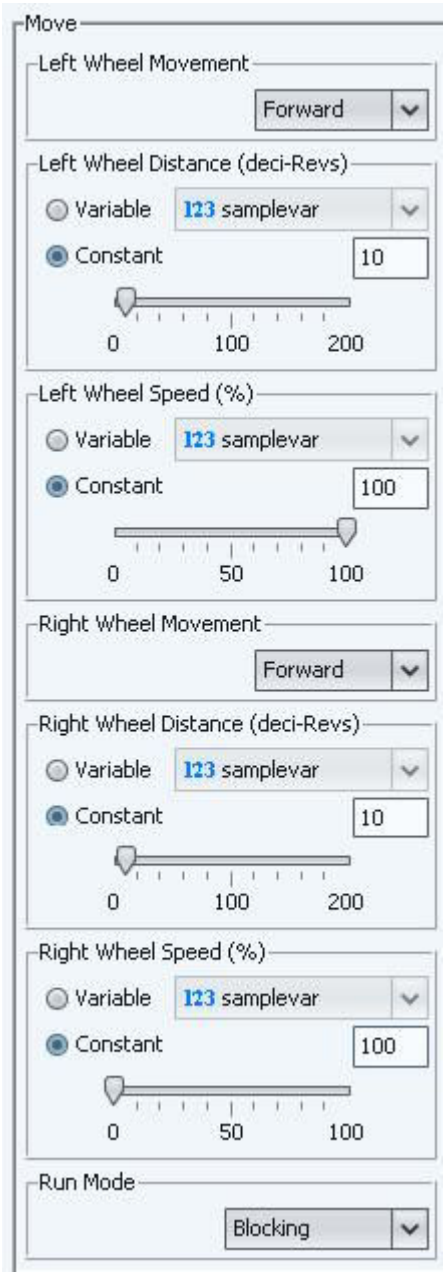
The delay tool is used when the robot needs to pause for a set amount of time. The delay is measured in milliseconds, or thousandths of a second. 1000 milliseconds equals 1 second. The Delay duration can be set by a fixed value (a constant), or by an integer variable value.





Move:

The move tool is used to make the robot motors turn. Each motor is controlled independently.



Settings for Left Wheel Movement and Right Wheel Movement control how the corresponding wheel will turn.

- Forward and Backward spin the wheels at the specified speed until the specified distance has been traversed.
- Brake will lock the wheel in place until a different Move instruction is issued.
- None causes the wheel to stop, but it does not lock.

The Left Wheel Distance and Right Wheel Distance can be set by the tool slider or by a variable to control how far the robot will go, in terms of 1/10 motor revolutions (deci-revolutions). The maximum is 200 deci-revolutions.

The Left Wheel Speed and Right Wheel Speed can be set by the tool slider or by a variable to control how quickly the wheels turn, where 100% indicates that they spin as fast as possible.

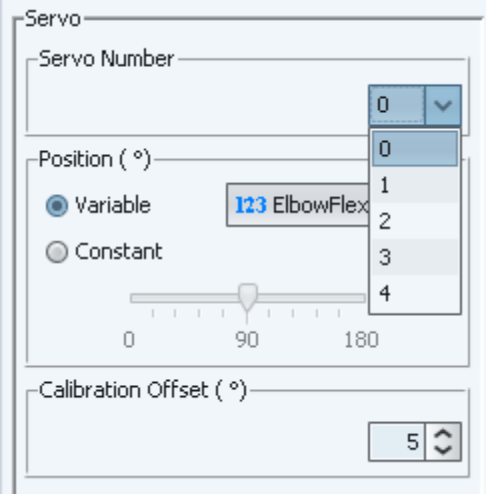
Finally, Run Mode provides an additional modifier.

- The default mode is Blocking, which tells the program to wait until the motors have moved the desired distance. This mode would be used to precisely control the distance traveled
- Non-blocking mode tells the program to keep going to the next command even while the motion command is performed. This mode can be used along with an input command to avoid a collision or play a tune while moving.
- Freerunning mode behaves similarly to non-blocking, except the wheels will keep turning forever or until a new move command is performed.

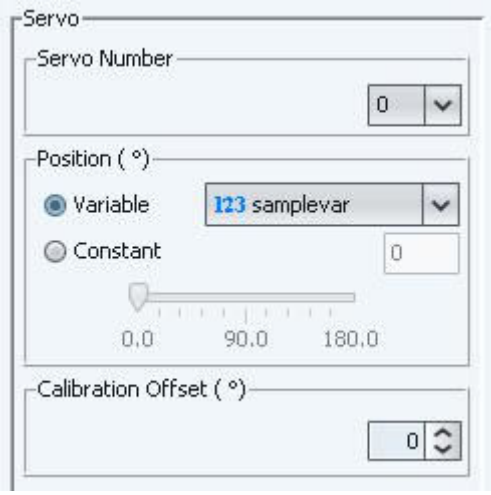
Servo:

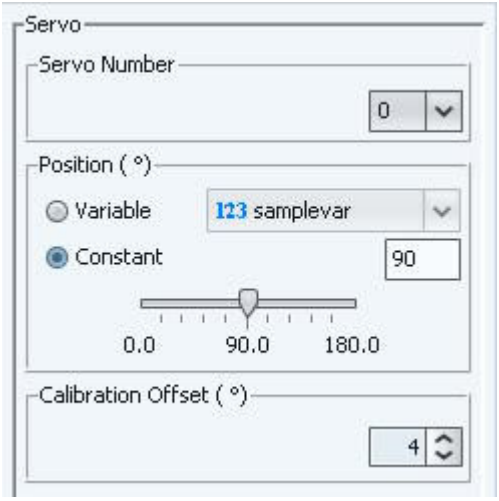
The Servo tool is used to control up to five servos connected to the controller board. A servo is a special kind of motor that does not rotate freely, and instead turns to a precise, repeatable rotation position like hands on a clock. Servos are used on radio control airplanes and cars to steer, fly and adjust throttle settings, and are also used on robotic arms.

Each Servo tool on the main screen can control one of five servos. A dropdown lets you pick the servo to control.



Servos are set to rotate from 0 to 180 degrees through use of a variable, or through a constant with the tool slider or the entry window.



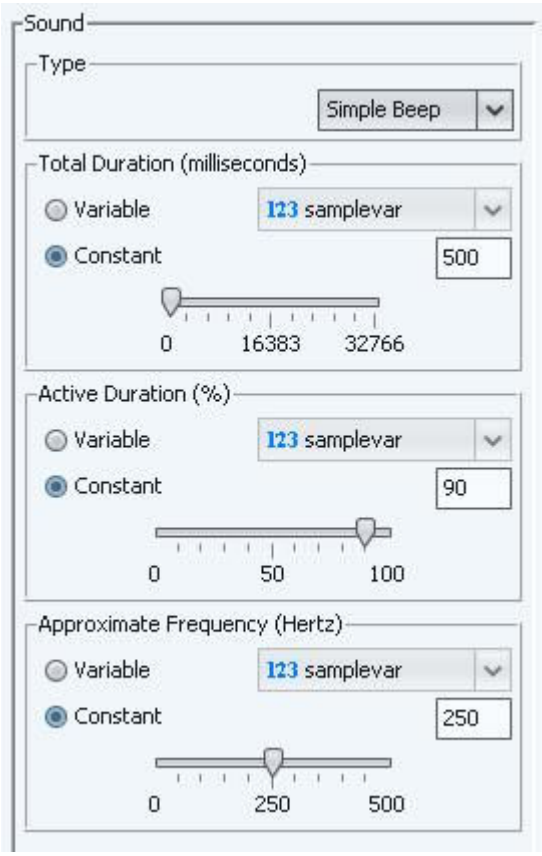


Since different models of servos do not rotate the same amount, a calibration offset allow you to tune CEENBoT Commander to get close to the 0 to 180 degree range. The Tool Description window describes how to calibrate your servos.

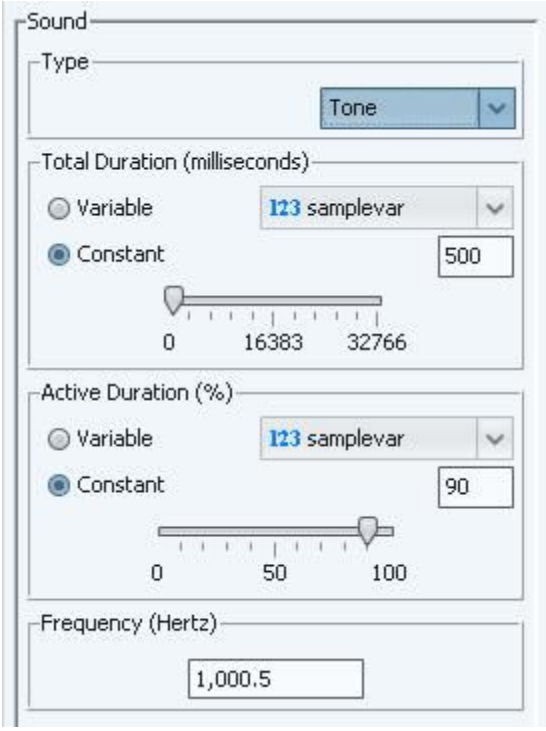
Sound:

The Sound tool produces tones through the on-board speaker using one of three types control: simple beeps, tones or notes. The sound event duration, time of the tone, and frequency of the tone all can be controlled. Sound event duration is the entire time spent on the tool, including the time sound is produced and an optional quiet time.

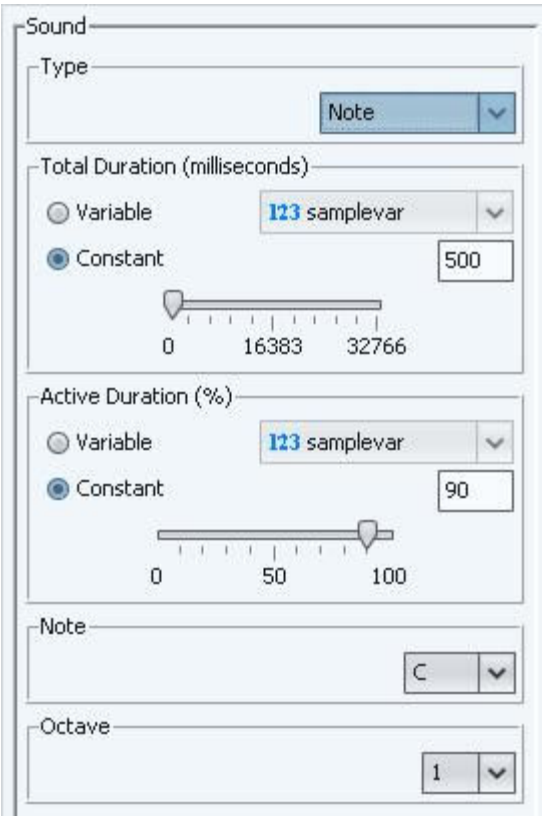
Simple Beep: If sound type is set to “Simple Beep”, the tool shows controls for sound event total duration, active duration (noise making time), and approximate frequency. Each control can be set using a variable, or with the tool slider. A simple beep might be used to signal to a user.



Tone: If sound type is set to “Tone”, the tool shows controls for sound event total duration, active duration (noise making time), and frequency. Duration controls can be set using a variable, or with the tool slider. The Frequency control is set in the tool control window to the desired frequency in Hertz (cycles per second). A tone might be used to send a more complex signal to a user, demonstrate the concept of frequency or Hertz, or play a simple tune.

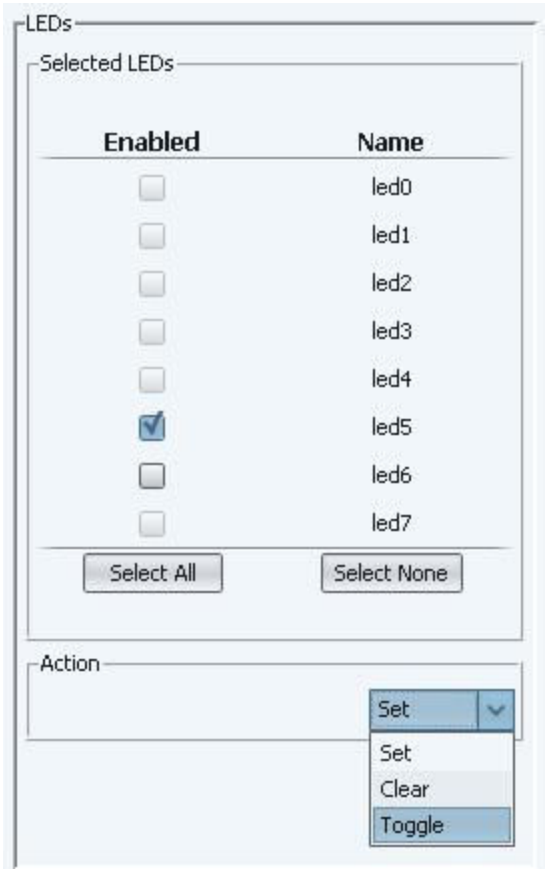


Note: If sound type is set to “Note”, the tool shows controls for sound event total duration, active duration (noise making time), and musical note and octave. Duration controls can be set using a variable, or with the tool slider. The Note and Octave control is set in the tool control window to the desired musical note. A note might be used to play a song.



LED:

The LED tool is used to control the LEDs. A check indicates which LEDs will be controlled. The available LEDs vary by platform, so more LEDs are listed than are on the controller board. The Action drop down allows the LEDs to be illuminated (set), dark (clear) or changed to the other state (toggle). Toggle is useful for flashing the LED when you don't know what its previous state was.



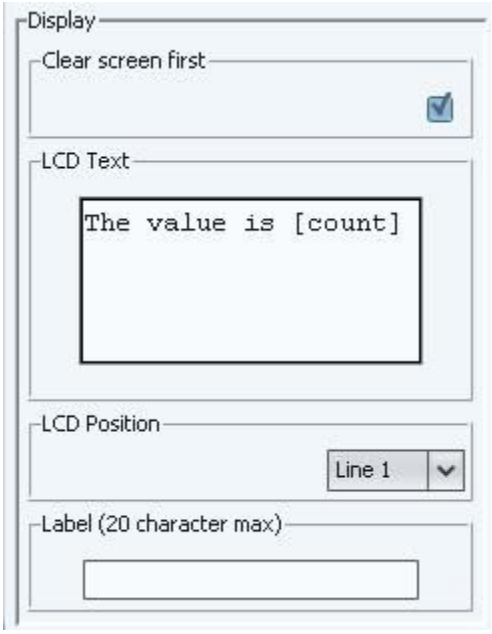
Platform	LED	Label	Color
324	led6	D3	Green
	led5	D2	Red

Display:

The display tool is used to show text on the LCD screen. The screen can be cleared or overwritten when the tool is performed, the line where the text appears can be controlled, and variables can be displayed on the screen.

The value of variables can be printed by entering the name of the variable with square brackets around it ("[" in front and a "]" after it). As an example: If the variable [COUNT] has been set to a value of 10, the display will show "The value is 10" on the first line of the LCD screen.

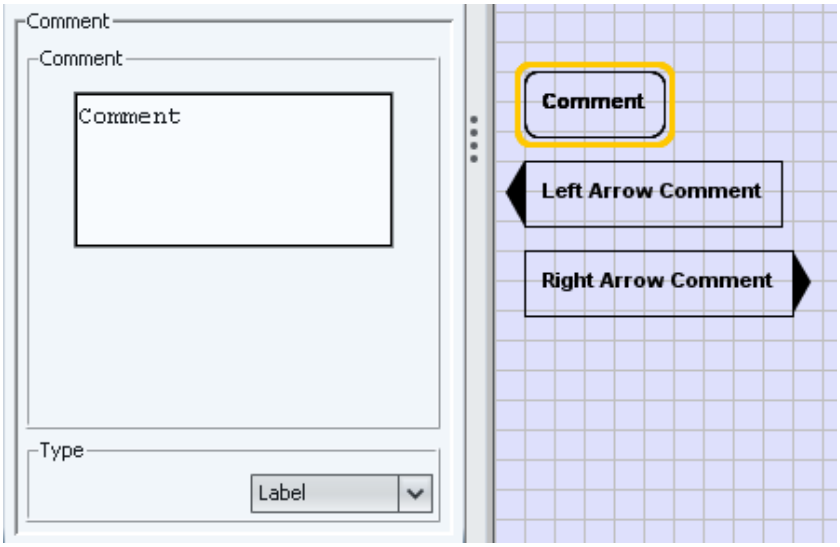
On the Commander Main Screen, display tools can show a label to help describe its purpose in the program.



Comment:

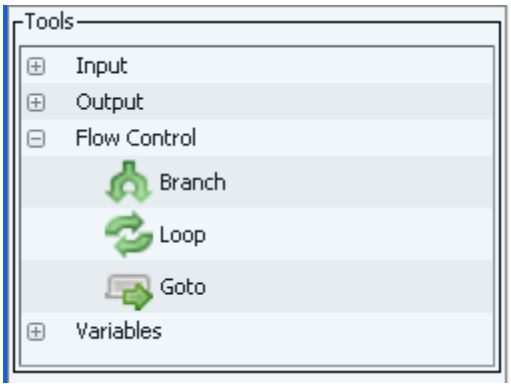
The Comment Tool is used to add user comments to the current program.

Comments do not affect the way a program runs, but do allow programmers to leave notes, hints and other messages within the visual program. Comments can be boxes with rounded corners, or have an arrow on the left side or the right side of the box.

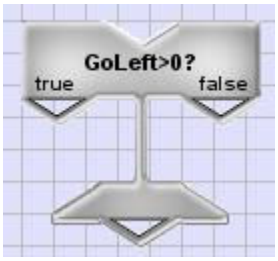


Comments can be used to remind yourself of a problem, question, or description of what a tool is doing.

Flow Control Tools

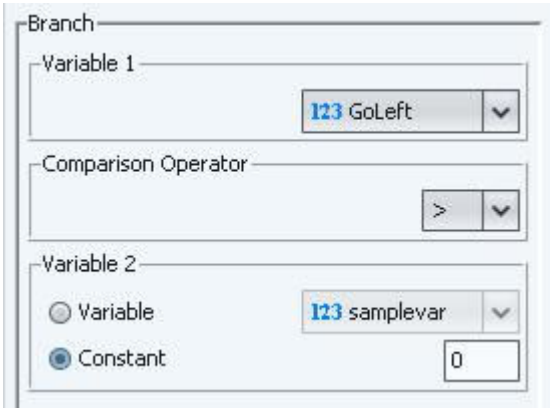


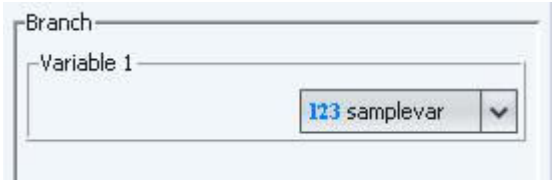
Branch:



The branch tool is used when you need a decision to be made based upon the value of a variable. Note that when comparing an integer with a constant value, the constant must be in the range -32768 up to 32767.

If the variable used by the branch is an integer, the tool window allows test conditions to be set. If the variable is a Boolean, no conditions are needed to set true or false.

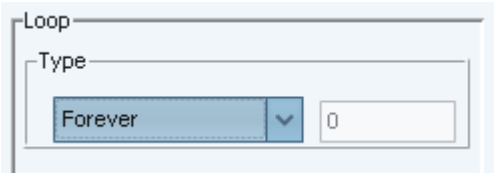


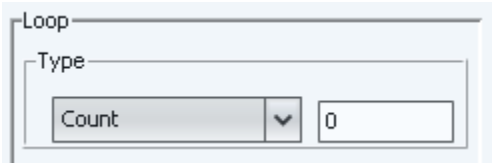


Loop:

The Loop Tool will repeat a chain of instructions for a controlled time or count. Three modes or types are supported: Forever, Count, and Conditional.

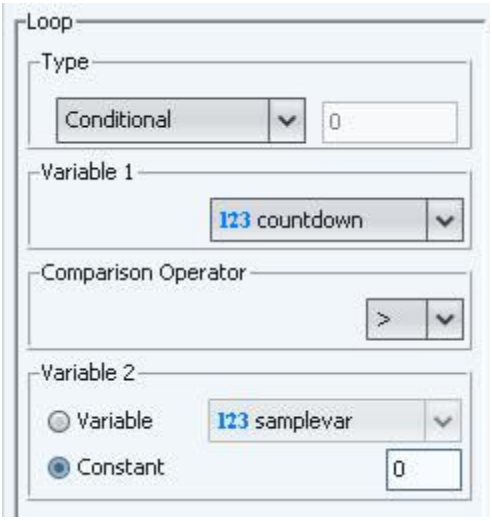
Execute the instructions in the loop continuously.

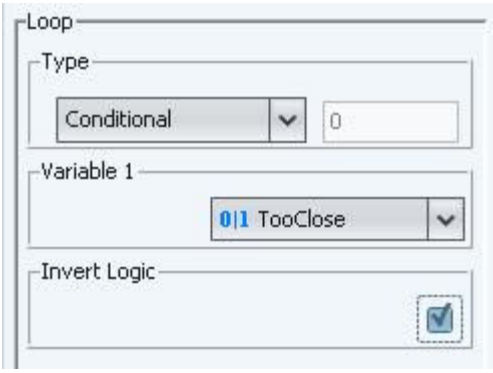




Count executes loop contents a set number of times or iterations. The number of iterations must be between 0 and 65535. Note that if you set the count to 0, it will not run at all.

Conditional executes loop contents until the value of a variable matches a preset condition. When comparing an integer with a constant value, the constant must be in the range of -32768 to 32767.



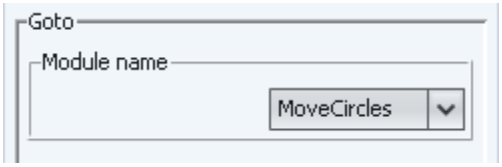


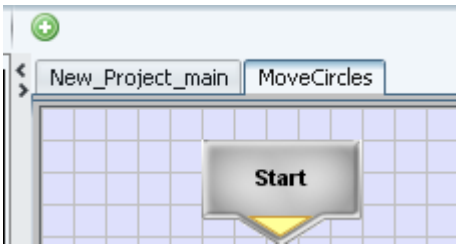
If the variable for this loop type is a Boolean type, the Invert Logic checkbox is available. If checked, a logical inversion (NOT) is performed on the variable before it is tested. This is called negative logic.

Goto Tool

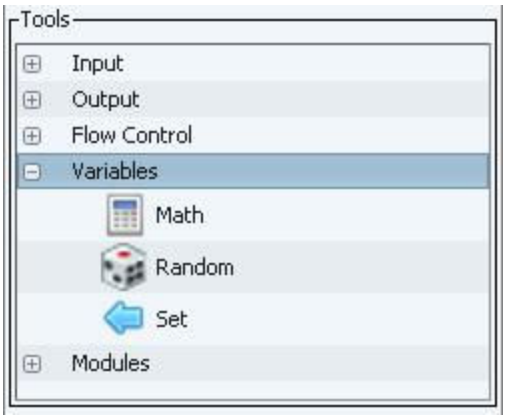
The Goto Tool is used to jump to another module in the current program. Once that module chain has been completed, the program flow resumes at the next tool.

To use the Goto tool, the module must already be created. The module can be changed at any time. Instructions for creating modules are on page 43.





Variables Tools

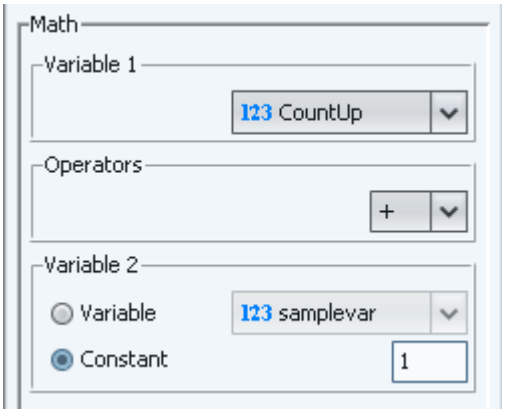


Math Tool:

The Math Tool is used to mathematically set the value of variables while the program is running.

For example, selecting “+” as an operator and setting Variable 2 to a constant with a value of 1 will cause the value of "Variable 1" to increase by 1 each time the tool is run

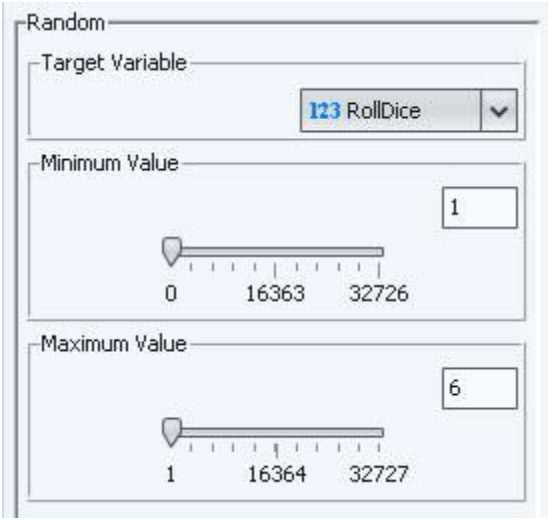
Note that variables must not exceed the range of -32768 to 32767.



Random Tool:

The Random Tool allows you to set a variable to an uncertain value between a minimum and maximum value.

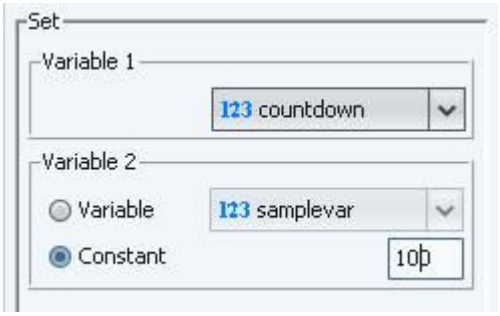
- Select the variable you want use to hold the random value in the "Target Variable" dropdown box.
- Set the "Minimum" and the "Maximum" values to the range of values you want returned. If your "Minimum" value is larger than your "Maximum" value, the robot will switch the two values and pick a result between the two.

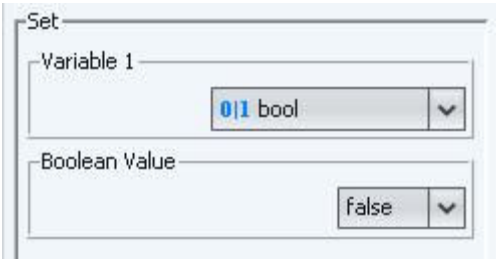


Set Tool:

The Set Tool allows values to be assigned to variables.

Values used in the Set Tool must fall in the range of -32768 to 32767 for integers, and true or false for Boolean types.





CEENBoT Commander Programming Tool Quick Reference

Start is a special tool that CEENBoT Commander creates on all programs. All user variables are defined here, either as integer or Boolean types.

Input

 Proximity - Boolean (True / False) assign to a user variable

 Switch - Boolean (True / False) assign to a user variable

 Remote - Remote controller input assigned to system variables.

Output

 Delay - Wait for a while – set through slider, number or variable

 Move - Turn motors and wheels forward, backward or brake – set speed and distance

 Servo - Rotate servo arms – set servo 0-5 and rotation in degrees or microseconds

 Sound - Make sound on speaker with beep, tone or note – set length, pitch or note

 LEDs - Illuminate red and green LED – set LED and on or off

 Display - Show text on LCD screen

 Comment - Make a note on the work space

Flow Control

 Branch - Control path of program execution based on a variable

 Loop - Repeat a part of the program until a condition is met

 Goto - Execution to another program (a module) and return back

Variables

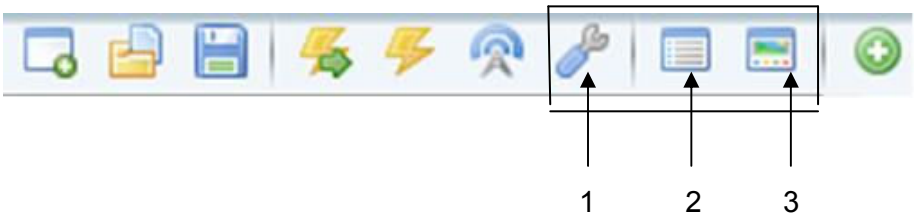
 Math - Add, subtract, multiply and divide, and put the results in a variable

 Random - Generate a “random” number between a low and high number

 Set - Put a numeric value into a variable

Option and Code View Screens:

CEENBoT Commander let you change the appearance of the program and switch the view of programs you have created from the visual style to their native programming language, C.



1. Options
2. View C Code
3. View Graphically

Options

The options tab lets you customize CEENBoT Commander for different looks, robot types, and tools available to use.

Change Target Robot Type.

This option allows choice of Board Type, Revision and Port. CEENBoT Commander will be used for other robot types, but for now only the AVR robot is supported.

You should not need to change these settings.

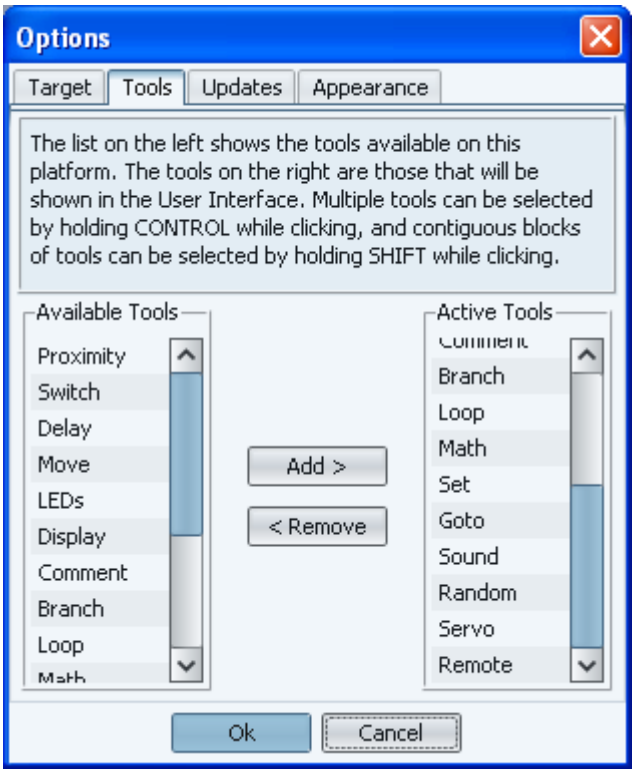
Enable or Disable Tools

CEENBoT robots can have a number of accessories like servos installed. Also, in some cases, it may be helpful to hide tools that are new or unfamiliar. This option allows selection or removal of tools that will appear in the Tool Box

To add a tool, highlight it on the left side and click <Add>

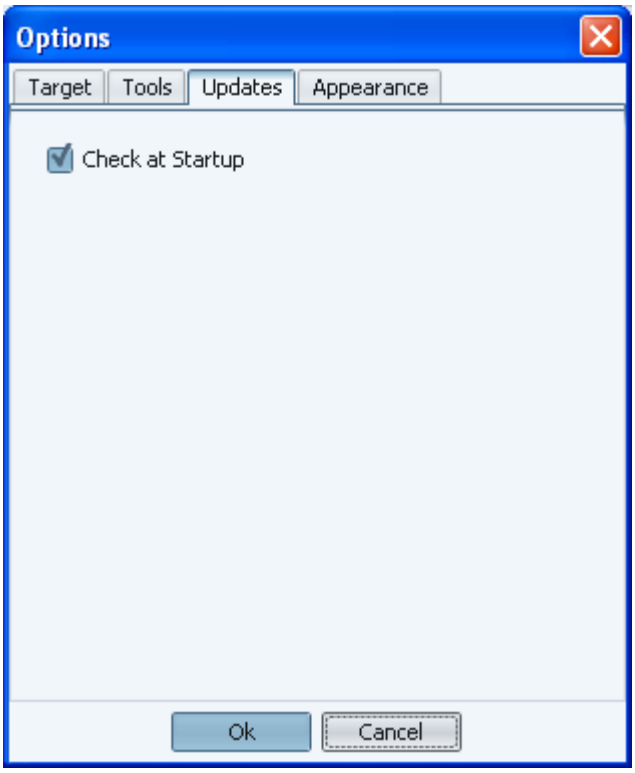
To remove a tool, highlight it on the right side and click <Remove>

Click <OK> to record your change



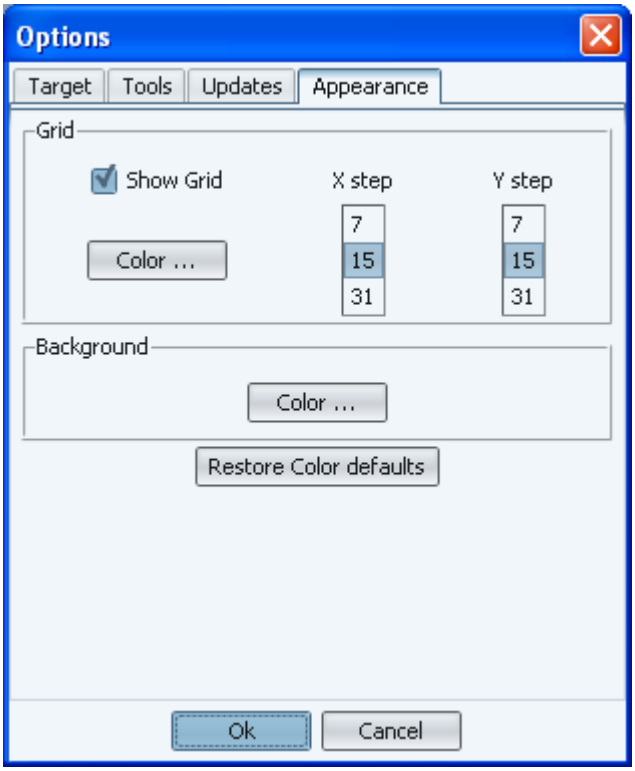
Automatic Check for Updates.

CEENBoT Commander now includes a feature that at startup, checks the Internet to see if new versions of the program and other files are available. You can enable or disable this check at startup on the “Updates” tab.



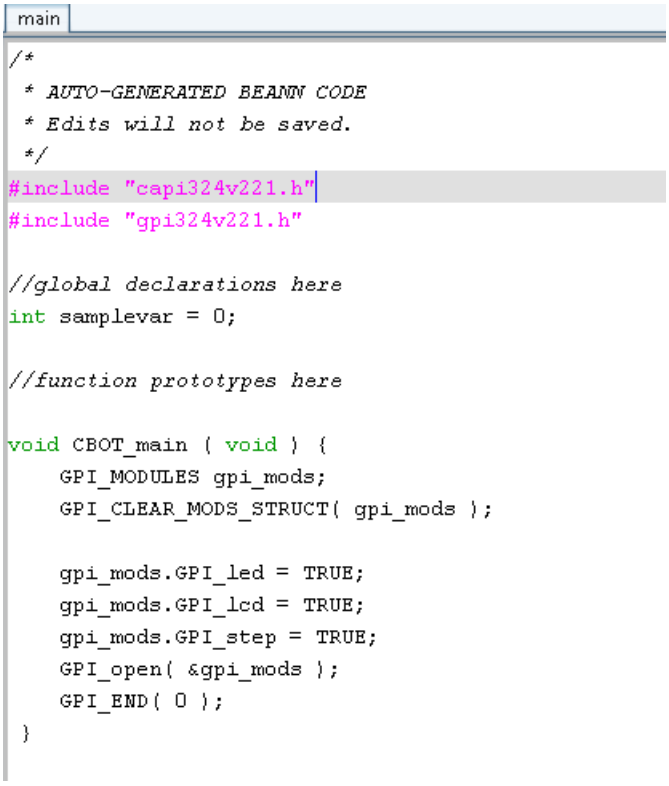
Change Appearance.

CEENBoT Commander's visual appearance can be changed to your tastes. The size and color of the main screen grid can be adjusted and the background color of the screen can also be changed.



C Code Screen:

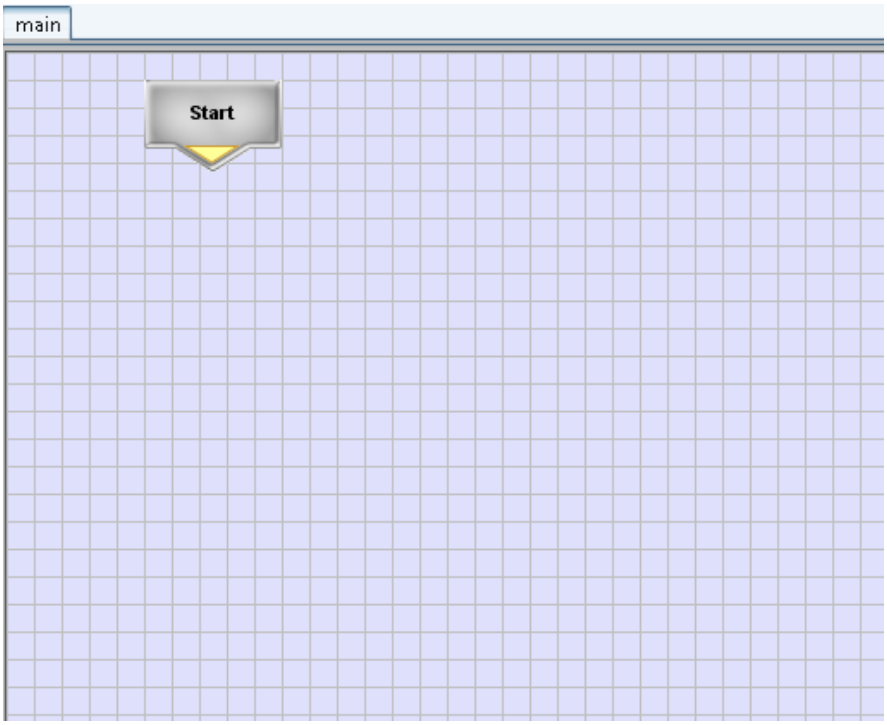
Allows the programmer to see what the visual programming will translate into when the program is prepared for building. CEENBoT Commander converts your visual programming into the C programming language before building it and transferring it to the robot. You may find the C language version of your program an interesting look under the covers. It is not necessary to understand this screen to use CEENBoT Commander. You cannot make changes on this screen.


main

```
/*
 * AUTO-GENERATED BEANN CODE
 * Edits will not be saved.
 */
#include "capi324v221.h"
#include "gpi324v221.h"

//global declarations here
int samplevar = 0;

//function prototypes here

void CBOT_main ( void ) {
    GPI_MODULES gpi_mods;
    GPI_CLEAR_MODS_STRUCT( gpi_mods );

    gpi_mods.GPI_led = TRUE;
    gpi_mods.GPI_lcd = TRUE;
    gpi_mods.GPI_step = TRUE;
    GPI_open( &gpi_mods );
    GPI_END( 0 );
 }
```

View Graphically:

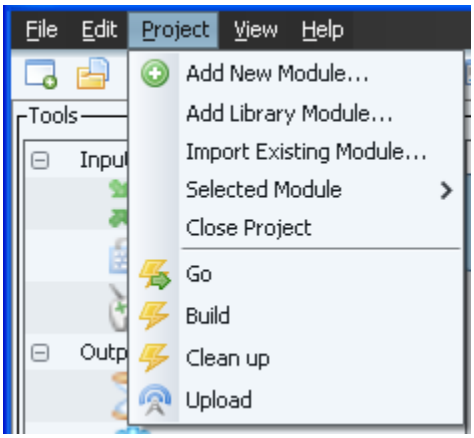
This icon button allows the programmer to return to the graphic screen after viewing the C code screen.



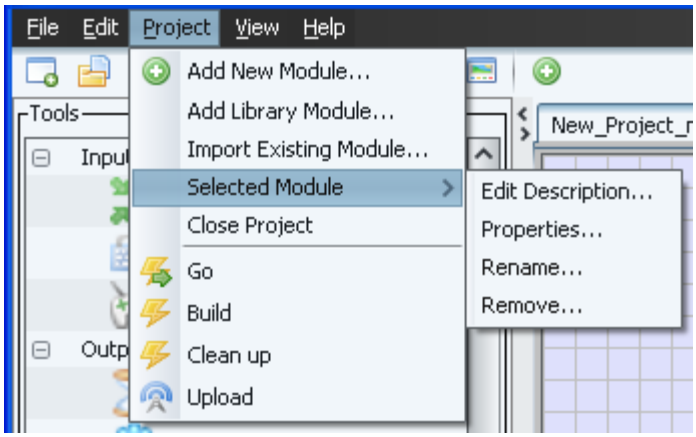
Drop Down Menu

At the top of the CEENBoT Commander window is a menu of choices, many of which are also available in the Icon Bar. Some unique features are under the “Projects” and “Help” tabs.

Projects Dropdown

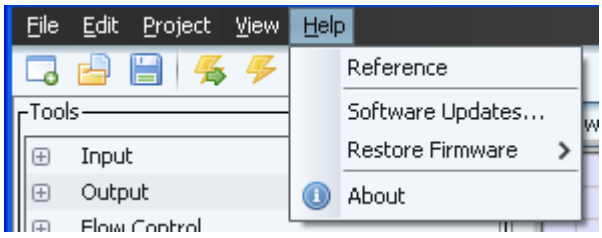


The “Projects” dropdown contains selections that allow creation of a new module, adding a library module (an already written module and included with CEENBoT Commander), import modules from other sources,



edit a module already part of this project, or close a project.

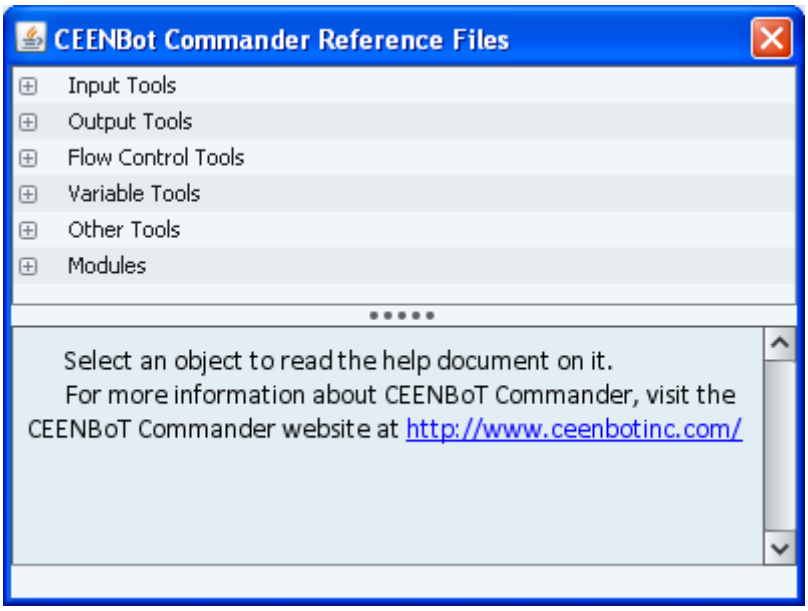
Help Dropdown



The “Help” dropdown contains some options that are important to using CEENBoT Commander.

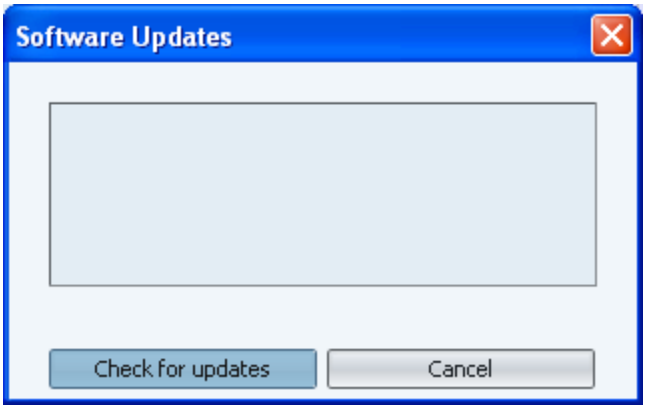
Reference

All of the tool descriptions are also available under “Help” > “References”



Manual Check for Updates

CEENBoT Commander will check the Internet for new versions of the program and other files. You can run this check manually by clicking “Help” > “Software Updates”.



Restoring CEENBoT Firmware

The CEENBoT Factory Base firmware is programming that provides basic functionality and battery charging. All CEENBoTs are shipped with factory firmware, so this allows you to return your CEENBoT to it's original programming.

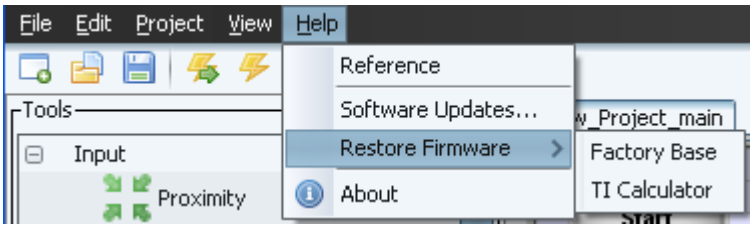
After programming your robot with CEENBoT Commander, you may want to restore the CEENBoT to the Factory Base firmware. This will be important if you need to charge the battery on your CEENBoT.

Other uses for restoring firmware including loading other people's CEENBoT programs, such as the graphing calculator control firmware. Also, occasionally, we add features or correct problems with factory firmware, and you can update your robot using this restore function.

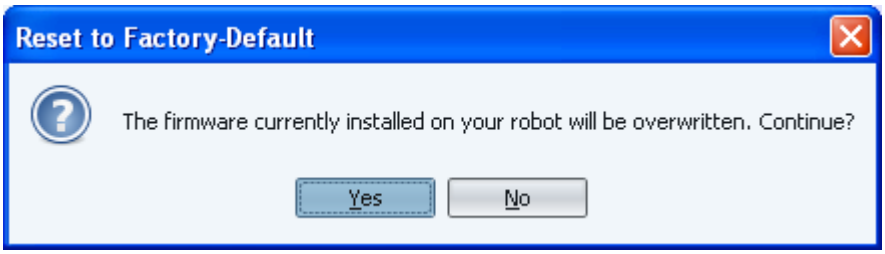
Pre-Upload Checklist

- Be sure to save your current project before restoring firmware onto to the robot.
- Be sure the robot wheels are not touching the floor or table top.
- Be sure the robot is powered on.
- Be sure the AVR Programmer is properly connected to both the computer and robot.
- Be sure the LED closest to the robot on the AVR Programmer is green.

When everything is ready: Select Help menu > Restore Firmware > Factory Base.



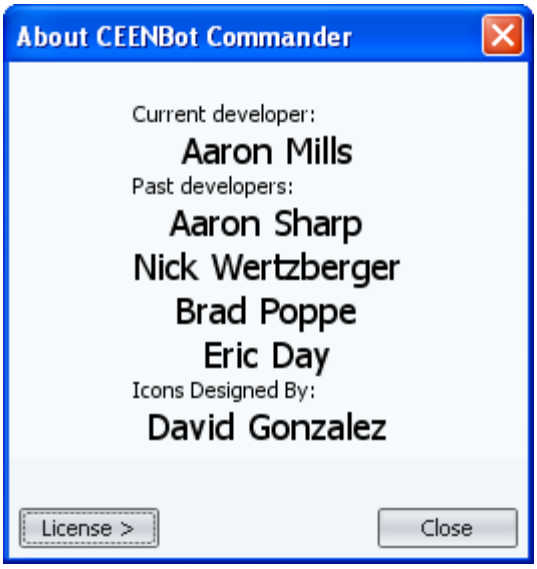
CEENBoT Commander will ask you to confirm that you want to do this.



After the firmware is uploaded, the program will start running immediately. Be careful the robot doesn't drive off of the table. The CEENBoT can now be used in the basic control mode and the battery can be charged.

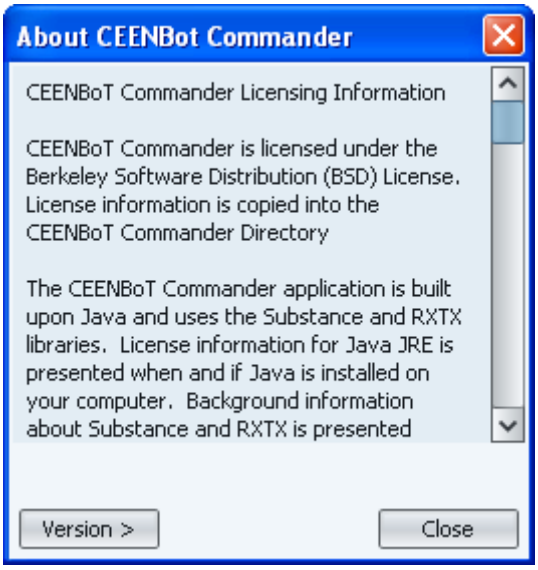
About

Information about CEENBoT Commander is available under “Help” > “About” tab, including versions, release dates,



The programming team and,

software licensing details.



Section 3 – Robot Programming Challenges

- **Basic Challenges**
- **Movement**
- **Geometric Shapes**
- **Loops**
- **Modules**
- **Problem Solving**

Programming Challenges

A great way to learn programming is to solve a problem with your program. Below are some ideas for challenges that will increase your understanding of robot programming. Take the challenge and create and test programs to perform the following tasks.

Because the robot will start running the program as soon as power is turned on, the robot is reprogrammed, or the robot is reset, each challenge starts with a 1 second delay. If your challenge includes motion, be sure to clear space on the floor or tabletop for it.

Basic Challenges

Display a Message:

1. Wait 1 second.
2. Display the message "Hello World!" on the LCD display.
3. Wait 1 second.
4. Clear the display.

Flash an LED:

1. Wait 1 second.
2. Turn on the green LED.
3. Wait 1 second.
4. Turn off the green LED.

Forward Motion:

1. Wait 1 second.
2. Move the robot forward two wheel revolutions.

Spinning Motion:

1. Wait 1 second.
2. Spin the robot 180 degrees. Spinning occurs when the wheels turn in opposite directions at the same time.

Move In a Circle:

1. Wait 1 second.

2. Have the robot follow a circular path. The robot travels in a circle when one wheel turns more slowly than the other.
3. Have the robot make a complete circle. This may take trial and error or calculation to travel 360 degrees.

Move Forward, Return:

1. Wait 1 second.
2. Move the robot forward two wheel revolutions.
3. Spin the robot 180 degrees.
4. Move the robot forward two wheel revolutions.

Move In a Square, Spin:

1. Wait 1 second.
2. Have the robot make a square as it moves. One way to think about the motion needed to make a square is moving straight, then turning 90 degrees, and repeating these steps three more times.
3. Have the robot spin 360 degrees when it completes the square.

Combining Tools to Do More Complex Things

Delay, LED, and Move Tools

Remember that programs start immediately after being downloaded, or after the CEENBoT power is turned on, or after the reset button is pressed. You may wish to include a delay at the start of your program so the CEENBoT will wait a second or two before starting to move.

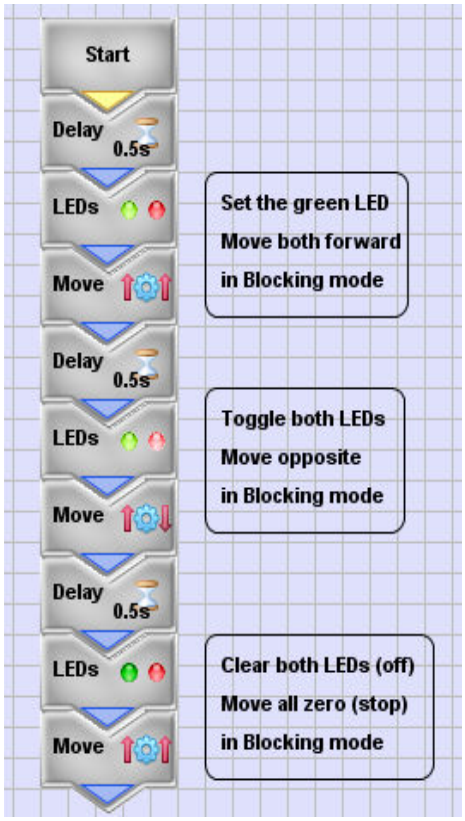
The LED tool can be used to provide visual feedback of the program. You can set (turn on), clear (turn off), or toggle (change) the red or green LEDs located near the reset button.

The Move tool is one of the more complicated tools since it has many different options and modes. You can set the direction, distance, and speed of each motor. You can also specify the run mode as blocking, non-blocking, or freerunning.

Use the Move tool blocking mode when you want the move command to use the full distance specified, like if you want to make a geometric shape or solve a maze. The sample program below shows movement with the blocking mode and use of delays and LED changes.

Program Description

1) Start with no variables defined
2) Delay 500 milliseconds
3) Set the green LED
4) Move both motors forward 30 deci-revs with a speed of 50 and use the blocking mode
5) Delay 500 milliseconds
6) Toggle both LEDs to turn off green and turn on red
7) Move motors in opposite directions 30 deci-revs with a speed of 50 using blocking mode
8) Delay 500 milliseconds
9) Clear both LEDs to turn them off
10) Move both motors forward 0 deci-revs with 0 speed to stop the robot even if using non-blocking or freerunning



Program Challenges

1. Create and test this program using the program description.

2. Modify the program so the second move command will turn the robot around 180 degrees so that it faces you.

3. Modify the program by adding and modifying move commands at the end so that the robot will return to the starting point and turn to face the starting direction.

4. Create a new program that will instruct the robot to trace a geometric shape like a triangle or a square. You could use masking tape to layout the shape on the floor and try to follow the tape.

5. Create a program to flash one of the LEDs using the Morris Code for SOS, three short flashes, three long flashes, and then three short flashes again. The robot could also move about frantically.

Non-blocking Movement

When using the Move tool, you can set the direction, distance, and speed of each motor, and specify the run mode as blocking, non-blocking, or freerunning. The blocking mode is used when you want the move commands to use the full distance specified.

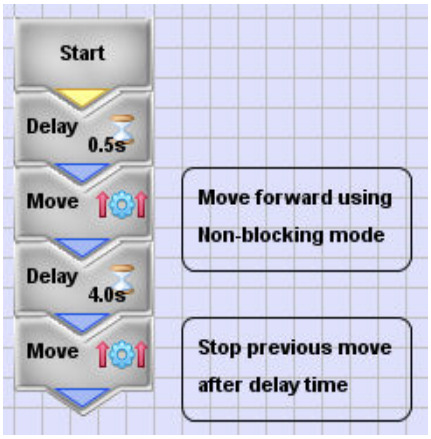
Use the non-blocking or freerunning modes when you want other commands that follow the move command to execute immediately. In non-blocking mode you enter the speed and distance; in freerunning mode you enter only the speed.

A non-blocking or freerunning move command will operate until another move command occurs. A delay command following a non-blocking move will cause the delay time to control how long the move command operates. A move command after the delay acts after the delay time is done.

The non-blocking move allows you to enter a limiting distance so if the delay time is too long, the move command eventually stops, the freerunning move runs forever. The sample program below shows a non-blocking move command that is controlled by a delay command.

Program Description

1. Start with no variables defined
2. Delay 500 milliseconds
3. Move both motors forward 40 deci-revs with a speed of 50 using the non-blocking mode
4. Delay 1000 milliseconds
5. With a Move command, set both motors to movement of none.



Program Challenges

1. Create and test this program using the program description.
   How far does the robot move? Measure and record your findings in the table below.

2. Modify the delay between the moves to 2000 milliseconds.
   How far does the robot move now? Measure and record your findings in the table below.

3. Continue to modify the delay between the move commands to complete the table. Make a time (X) and distance (Y) graph.

Time (s)	Distance (m)
0	0
1	
2	
3	
4	
5	

Programming Tasks Using Loops and Modules

Often when programming, the program will need to do something repeatedly forever, or until all work is done, or until a condition is met. Loops are one way that programmers handle this need.

In other cases, the same sequence of steps will need to be performed in different places of the same program. Modules let the programmer write the repeated steps once and use them over and over.

Loops can be used in a main program or included in a module. Below are examples of how you can use modules and loops types to solve some programming tasks.

Loop Until a Condition is Met - Wait for a button press

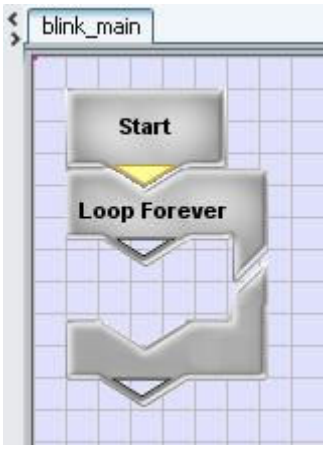
You may want to put a step in your program to make the robot wait until you push a button before it starts moving. This step can be used in place of waiting 1 second at the start of each program.

Using a variable, a loop tool, and a switch tool, how can you create a step that waits for a button to be pressed before continuing?

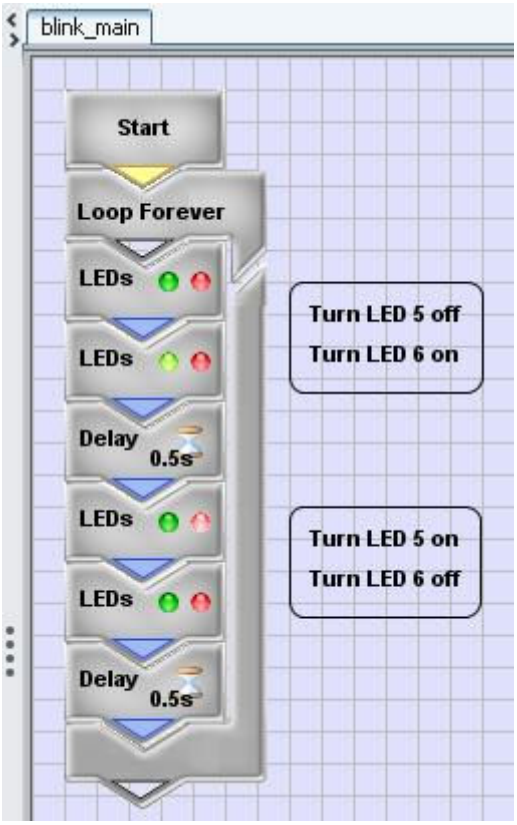
Loop Forever – Blinking LEDs

Use a loop to blink the LED lights forever.

Add a Loop tool set to run forever.



- Add LED and Delay commands to the program. Note that each LED command takes a separate tool to turn it on and off.



When you build and download your program, the robot flashes its red and green LEDs on and off until you stop it.

Counting Loops

A counting loop will use the Goto modules a certain number of times. Add an example of a counting loop

Modules

A module is a program within the program. It has the same structure as a regular program including the Start tool and other tools. What makes it different is that it can be used again and again in different places.

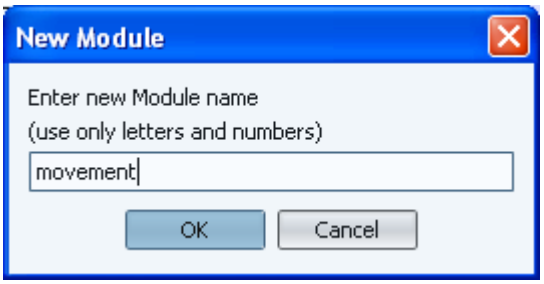
Using Modules to Move

One use of modules is to repeatedly move the robot a distance using fewer commands.

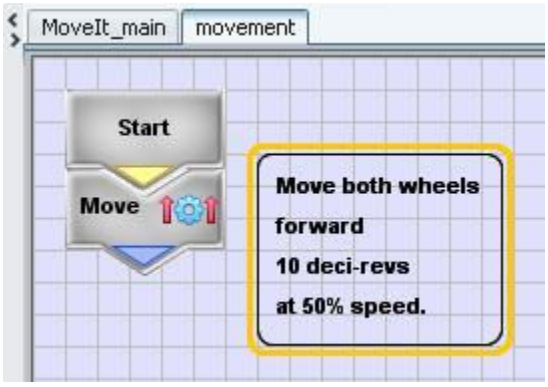
First create a module called **"movement"**.

- From the Icon Bar, click the green "+" icon to add a new module to the current project.

- Enter the Module name: **"movement"**.

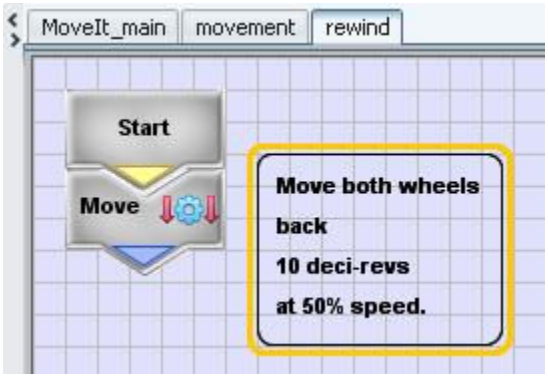


- Add a Move command to the **"movement"** module. Set the move parameters to move forward 10 deci-revs at 50% speed on both wheels.

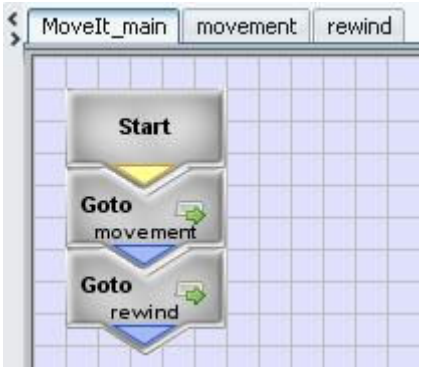


- Add a new module to current project. Enter the Module name **"rewind"**.

- Add a Move command to the **"rewind"** module. Set the Move parameters move in reverse 10 deci-revs at 50% speed on both wheels.



- In the main program, put a GoTo tool to **"movement"**, and then a GoTo tool to **"rewind"**.



When you build and download your program, the robot will roll forward one revolution, and then back to where it started.

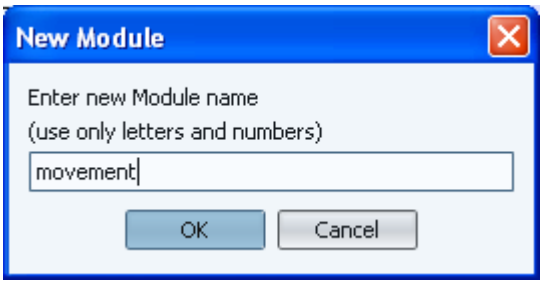
Using Modules to Make a "U" Turn

One use of counting loops is to make geometric shapes with fewer commands. This task will use two modules to turn your robot to face the other direction.

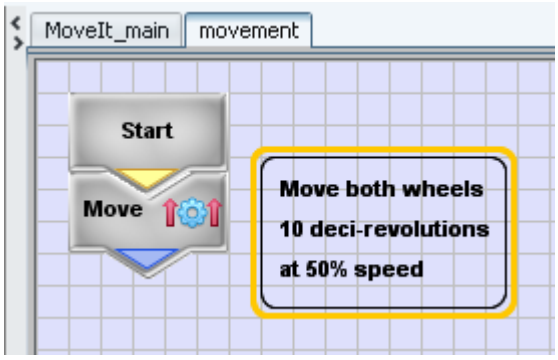
First create a module called **"movement"**.

- From the Icon Bar, click the green "+" icon to add a new module to the current project.

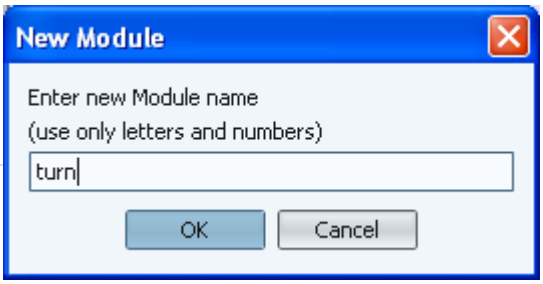
- Enter the Module name: **"movement"**.



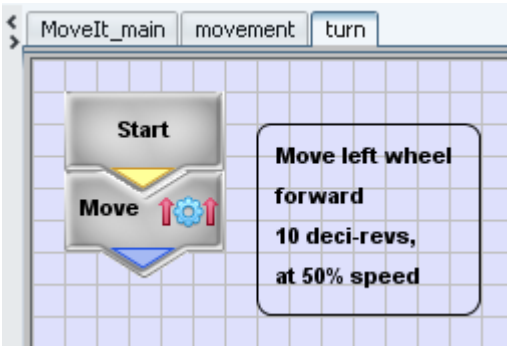
- Add a Move command to the **"movement"** module. Set the move parameters to move forward 10 deci-revs at 50% speed on both wheels.



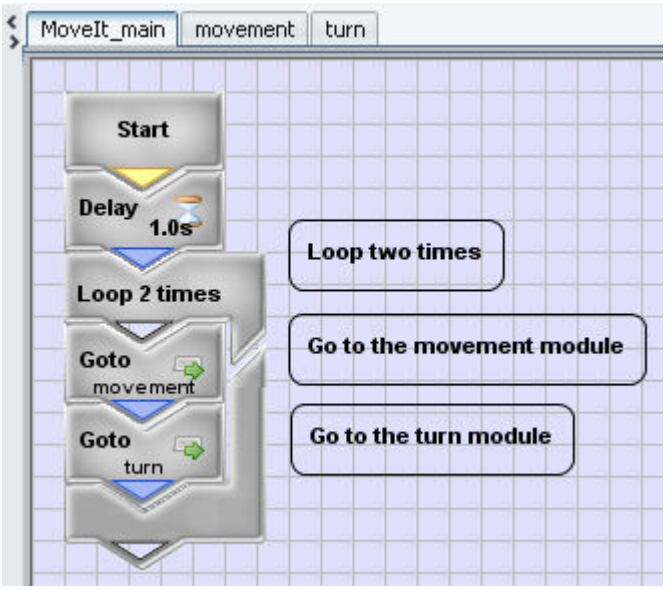
- Add a new chain to current project. Enter the Module name: **"turn"**.



- Add a Move command to the **"turn"** module. Set the Move parameters to forward 10 deci-revs at 50% speed on the left wheel.



- Add the Delay, Loop, and Goto modules to the main module. Set the Loop to the Count type that will run 2 times. Set the Goto commands to use the **"movement"** and **"turn"** modules.



Advanced Challenges

Challenge - Robot Geometry

What geometric shape can you make?
Triangle? Square? Star? Circle? Figure Eight?

Challenge - Robot Snow Plow

Create a program to clear the "driveway" of "snow".
The "driveway" is an area 3 feet by 5 feet that is taped off on the floor.
For "snow" use wads of paper or other objects that are randomly added placed on the "driveway".
You may choose the location to start and when to stop your robot "snow plow".

Hint: If you use paper wads as snow, you may notice the snow gets stuck under the wheels.
You may need to modify the front of the CEENBoT to better push the "snow",
or you can program the CEENBoT to plow the driveway backwards.

Terms and Acronyms

Atmel – Integrated circuit manufacturer of the microcontrollers used on the CEENBoT.

AVR - The AVR family of microcontrollers is manufactured by Atmel. It is one of the first microcontroller families to use on-chip flash memory for program storage. The CEENBoT controller board contains two of these microcontrollers

AVRISP mkII Programmer – A device manufactured by Atmel to write programs into the memory of an Atmel microcontroller. It connects to a PC or Mac type computer with a USB cable, and to the robot controller with a 6 pin cable.

CEENBoT – An educational robot designed and distributed by the University of Nebraska, Omaha Campus Computer & Electronics Engineering

CEENBoT Commander - A Graphical Programming Interface (GPI) program that enables drag and drop visual programming of the CEENBoT. The visual program is turned into a microcontroller program by Commander, and loaded onto the microcontroller with the AVRISP mkII programmer.

Firmware – The computer program loaded onto a microcontroller. It is unchanged when power is turned off.

Flash Memory – A type of electronic circuit that stores binary data for a microprocessor. The data in flash memory is non-volatile, or unchanged when power is turned off. Microcontroller firmware is stored in flash memory, and USB thumb drives are based on flash memory.

Infrared Sensors: A device that emits infrared light and detects reflection of the infrared beam from objects in front of the robot. A common use is to avoid collisions with objects or walls.
ISP - In-circuit Serial Programmer - Supports direct programming of AVR microcontrollers. The AVRISP mkII Programmer is an example of this.

LCD - Liquid Crystal Display - A digital display that uses an array of liquid crystal cells that change reflectivity based on the applied voltage. Images and text are presented on the display to be viewed and read.

Memory – Electronic circuits that store binary data for use by a microprocessor. The data in non-volatile memory is retained when power is turned off, and is used for program storage. RAM memory loses its data when power is turned off, and is used for temporary data storage.

Microcontroller – An integrated circuit containing nearly all parts of a complete computer system, including the microprocessor, non-volatile memory, RAM memory, and input/output connections.

PC Board, PCB - Printed Circuit Board - A sandwich of fiberglass and resin covered with copper sheet. Conductive traces are made using an etching process and mounting holes are drilled through the board. Integrated circuits and other electronic components are attached by soldering.

Solder – A metal alloy that melts at a low temperature. It is used to create an electrical and mechanical connection between wires, PC boards, electronic components and integrated circuits.

Stepper Motors: A motor that rotates in short, consistent angular movements rather than turning continuously. The CEENBoT stepper motors turn 1.8° with each step.

USB - Universal Serial Bus - A standard for connecting external devices to a computer. It is used for communication, control and power. The AVRISP mkII programmer connects to a PC or Mac computer through a USB connection.