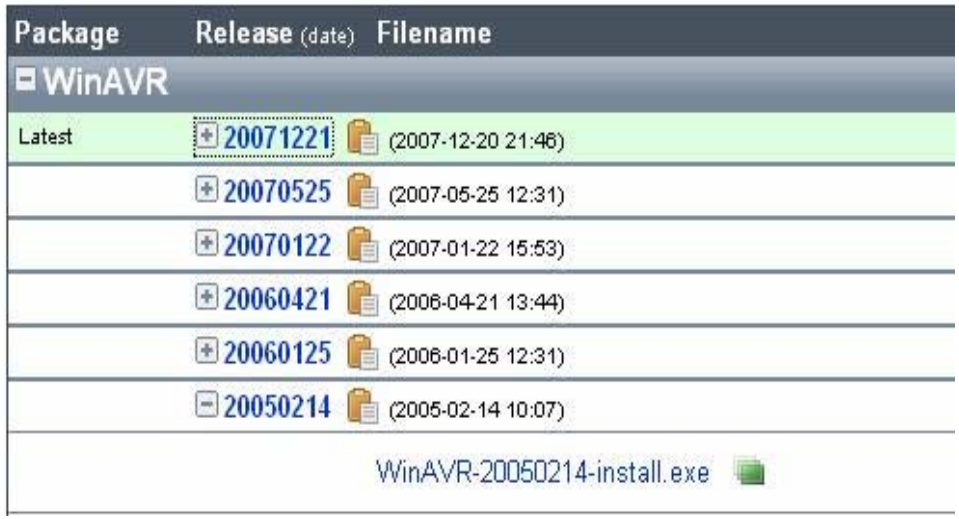
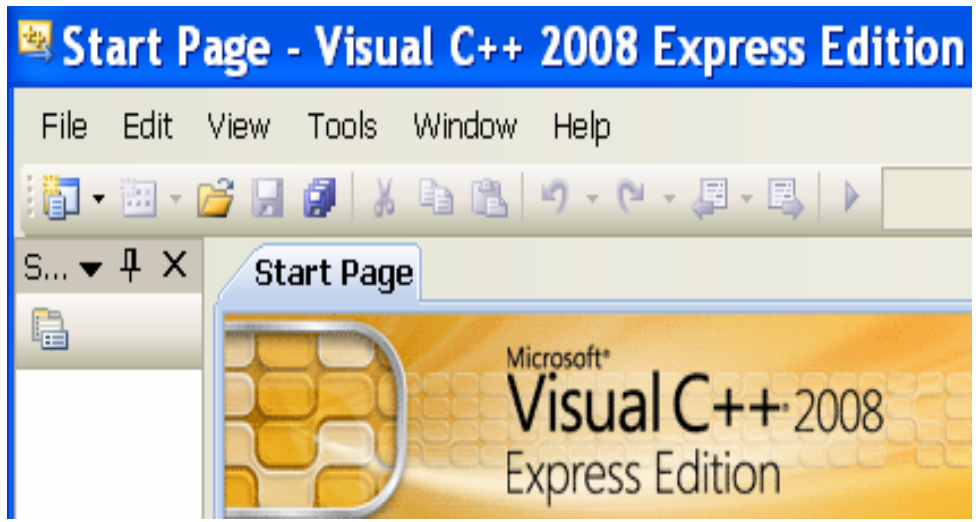
RoboPhilo Software Development Kit (SDK) User Guide

RoboPhilo Software Development Kit (SDK) User Guide





Chapter 1. RoboPhilo Software Development Kit (SDK) Overview

1.1 Overview

- User can add in about 4K program routines using 200 bytes of memory for variables in addition to the current RoboPhilo programs.
- The package provides API for user to develop c programs to interface with RoboPHILO robot.
- The package comes with the RoboPHILO firmware library (libPhilo.a, main.c, main.h, makefile) to add your routines to run inside RoboPhilo
  - User can download the free ATMEL's WinAVR compiler package to develop the c program with the SDK.
- The package comes with sample C++ program to send command to RoboPhilo over the serial cable
  - User can download the Microsoft Visual C++ 2008 Express compiler package to develop the pc program.
- The SDK provides API for interrupt handler, IO PORT, Servo Pose, and handheld remote interface for user to customize the program at varies level depending on the user's skill level.
- User can add functions to interface with the added hardware device for different actions.
- User can program RoboPHILO to perform the pre-downloaded motion by the GUI PHILO motion creator.
- User can program RoboPHILO to move to a pose defined or generated within the program.
- User can replace the default RoboPHILO functions with new functions and even the interrupt handler.
- User can download the new program to RoboPHILO using RS232 cable without special dongle hardware.
- With the SDK, user can turn the RoboPHILO into an autonomous robot interacting with additional object detection hardware
- The RoboPHILO SDK package is licensed to one RoboPhilo per serial number only. The programs will only work on the RoboPhilo with serial number submitted with the purchase.

Chapter 2. WinAVR Programming for RoboPhilo

2.1. Overview

2.1.1 The package comes with the following programs,

a. Makefile – to build the philo firmware
b. libPhilo.a – library for roboPhilo routines
c. Main.c – c program for user modification
d. Main.h – header file for roboPhilo routines

2.1.2 Infra Infra Red remote functions

a. getRemoteCmd()	- return key command entered with infra red remote
b. processRemoteCmd()	- process the action for the key command entered
c. remoteCmdDone()	- confirm the previous entered remote key is processed

2.1.3 IO functions for PORTA PIN 0 - 3, PORT D PIN 0 - 3

a. readPortA ()	- read PORTA data
b. writePortA ()	- write PORTA data
c. readPortD ()	- read PORTD data
d. writePortD ()	- write PORTD data

PortA pin 3 is used for setup and low power detection	- can be used as input with software changes
PortD pin 0 is used for Rx	- can be used as IO with software changes
PortD pin 1 is used for Tx	- can be used as IO with software changes
PortD pin 2 is used for Infra Red receiver	- can be used as IO with software changes

2.1.4 ADC functions for PORTA PIN 0 - 3

a. getAdc ()	- return the specified ADC channel data

PortA pin 3 is used for setup and low power detection	- ADC value is lower without hardware modification

2.1.5 RoboPhilo functions

a. runRemoteCmd()	- run the routines assigned to the handheld remote
b. getPose()	- get the RoboPhilo current servo positions and speed, mode, step
c. runPose()	- move all 24 servos at the specified speed, mode and steps
d. runServoPos()	- move one servo at the specified speed, mode and steps
e. waitN20ms()	- wait for the specified numbers of 20 msec, n20ms = 50 => 1 second
f. getServoATV()	- get the current atv value for the specified servo
g. setServoATV()	- set the atv value for the specified servo
h. getServoPower()	- return current servo power state. 1 = on, 0 = off
i. setServoPower()	- turn on/off the servo torque
j. toggleServoTorque()	- toggle on/off the servo torque

2.1.6 Command packet from pc functions

a. receiveChar ( )	- return a character read from the Serial port
b. isPacketAvailable()	- return true if a command packet is received from pc
c. processPacket()	- execute the command packet from pc
d. userProcessPacket()	- execute the user defined command packet from pc

2.1.7 Debug functions

a. sendChar()	- send the input character to the pc serial port
b. sendString()	- send the input string terminated by '\0' to the pc serial port
c. displayCharValue()	- send the input char value as 3 digit numeric character to the pc serial port
d. displayIntValue()	- send the input integer value as 5 digit numeric character to the pc serial port
e. displayStringValue()	- send n input char value as 3 digit numeric character separated with " : " to the pc serial port

Chapter 3. PC Based Programming for RoboPhilo

3.1. Overview

3.1.1 The package comes with the sample c++ program (pc.cpp) with the following routines source code,

- open and setup the serial com port
- setup the command packet to send to RoboPhilo
- routine to move one servo position
- routine to move all 24 servos pose
- routine to send user defined command executed by the RoboPhilo

User can use Visual C++ with the Platform SDK or Visual C++ 2008 Express to make changes and compile the program.

The compiled program can control the RoboPhilo running with the SDK compiled firmware only

3.1.2 Serial COM port functions

a. openComPort()	- open the com port to communicate with the roboPhilo
b. writeCom()	- write the data packet to roboPhilo
c. readCom()	- read the data packet from roboPhilo

3.1.3 Command packet functions

a. formatPacket()	- pack the global outPacket buffer with data, header and checksum to send
b. setOutCmd()	- pack the opcode, speed, mode and steps into the data buffer for short command
c. setOutFullCmd()	- pack the opcode, speed, mode and steps into the data buffer for 24 servos

3.1.4 Operations send to RoboPhilo

a. sendCommand()	- send the simple command to RoboPhilo
b. sendPose()	- send the full state pose command to RoboPhilo

Chapter 4. Samples

4.1 The package comes with sample routines

a. Show the user how to program and use the routines to perform the command inside the RoboPhilo
b. Show the user how to program and use the routines to send command to the RoboPhilo
c. User can comment out certain routines or copy them into the main program to try out

4.2 RoboPhilo firmware sample routines in main.c

a. initialization()	- show user how RoboPhilo is initialized
b. processRemoteCmd()	- show how a remote command is processed
c. userInitRoutine()	- let user customize run once actions - show how to call runServoPose(), runPose(), getServoATV(), getPose(), setServoATV(), waitN20ms(), runRemoteCmd(), setServoPower(), toggleServoTorque()
d. sampleAdc()	- show how to call getAdc() and display adc values
e. sampleReadPortA()	- show how to call readPortA() and display their values
f. sampleWritePortA()	- show how to call writePortA()
g. sampleReadPortD()	- show how to call readPortD() and display their values
i. sampleWritePortD()	- show how to call writePortD()
j. userLoopRoutine()	- let user customize show how to
k. userProcessPacket()	- show how to add a user define operation inside RoboPhilo

4.3 pc sample routines in pc.cpp

a. openComPort()	- show how to open and initialize a serial COM port
b. writeCom()	- show how to write the data packet to roboPhilo
c. readCom()	- show how to read the data packet from roboPhilo
d. formatPacket()	- show how to pack the command packet with sync byte and checksum
e. setOutCmd()	- show how to pack the short command data with opcode and parameters
f. setOutFullCmd()	- show how to pack the full command data with opcode and parameters
g. sendCommand()	- show how to send a simple command to RoboPhilo
h. sendPose()	- show how to send a full command to RoboPhilo
i. tmain()	- show how to call openComPort(), sendCommand() and sendPose()