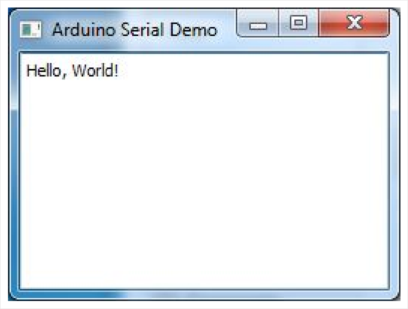
A slick threaded GUI for Arduino projects with Python and Qt

Posted on May 16, 2011 by admin in Uncategorized



Hello, World!

Having your Arduino communicate over the serial monitor is nice, but for longer running installations and apps, a GUI is a good way to go. This blog post shows how to use a threaded GUI client to listen to the Arduino. It uses Python and Qt, from Nokia.

This example is as simple as possible. It uses threading to control the GUI, and a queue to communicate between the Arduino and the GUI.

Tools

There are many GUI toolkits available, but I like Qt. It is world class, fully cross platform, and presents a professional looking result. Also, it has a cool system of signals to communicate between objects, avoiding callbacks. This means it is safe. If you are going to build professional GUI's with Python, in my opinion Qt is the way to go. This tutorial uses Qt4.7 and Python 2.7 (I imagine anything over Python 2.4 will work, but I haven't tried).

I'm a big fan of Python, because it is an easy general purpose language and doesn't get in the way of my brain while I'm problem solving. To get Qt and Python talking, we use PyQt. You can use PySide as well.

Setup

To get started, you need to install Python (Mac and Linux have it pre-installed), PyQt and pySerial. I like *easy_install*, a Python package installer. You might as well download Qt4 while you're at it – I recommend going the whole hog and getting the Qt Creator Binaries for your system. It includes Qt Designer, which is nifty.

The GUI

Next step is running the GUI application. Download ArduinoPyQt.txt and rename it to ArduinoPyQt.py. Before we go over what it does, run it from your IDE or *Python ArduinoPyQt.py* from the command line to check it works. It should bring up a window like the picture above, but blank. (I'm on Win7 – the skin will match whatever system you are on.) It is blank because we don't have the Arduino connected yet.

The Arduino sketch

We're going to use the most t

Chat with us

```
void setup() {
Serial.begin(115200);
}
void loop() {
Serial.println("I blinked");
delay(1000);
}
```

It just pushes "I blinked" down the serial port once every second. Couldn't be simpler! Connect your 'Duino and load that up using the Arduino IDE.

The last thing you need to do is to make sure that the serial port is correct. Go into the code and change SERIALPORT to whatever the correct port is for your 'Duino.

FREE SHIPPING TO THE USA FOR ORDERS OVER $10

PRODUCT CATEGORIES

New Products
Air Quality Egg
Display LED LCD
Kits and Assembled Projects
Microcontroller/Arduino
Parts and Accessories
Sensors
Shields and Add-ons
Wireless links

LATEST BLOG POSTS

Wildfire now available on Codebender.cc
Wicked Headphones
WildFire: FTDI Driver Issues on Mac OSX 10.9
WildFire support package update coming soon
CE mark = "China Export"?

TUTORIALS

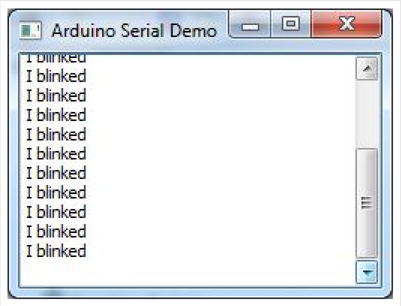
MintyTime
Day Counter
Nanode Gateway/Remote

FORUMS AND BLOG

Wicked Device Blog
Forums. Get help!

CONTACT US

Wicked Device LLC
1172 Ellis Hollow Rd
Ithaca, NY, 14850
email us



```
I blinked
I blinked
I blinked
I blinked
I blinked
I blinked
I blinked
I blinked
I blinked
I blinked
I blinked
```

Run the sketch again, and it should be saying “I blinked” once every second. You see it? Yay!

Tags: Arduino, GUI, PyQt, pySerial, Qt, threads

About admin

View all posts by admin →