

IR-Blue-duino

IR-blue-duino utilise un capteur thermique Melexis MLX90620 pour transformer un téléphone en caméra thermique

Inspiré du projet open source ir_blue project :

<http://www.rhworkshop.com/pages/open-source-ir-blue>

Fichiers originaux :

<https://github.com/RHWorkshop>

Portage sur la plateforme Arduino du programme trouvé ici :

<https://github.com/RHWorkshop/IR-Blue-Eagle-Files/tree/master/IR-Blue%20BT2/Firmware>

Le hardware original utilisait un PIC.

IR-blue est associé à une application sur téléphone (Android ou iPhone)

Le téléphone reçoit les données du MLX et les superpose à l'image de la caméra du téléphone

L'application pour le téléphone peut être trouvée ici:

<https://github.com/RHWorkshop>

Le présent logiciel n'a été testé qu'avec l'application Android

L'aide sur l'application peut être trouvée ici :

<http://www.mardaso.com/Products/FIRsensorFree/Firsensor.html>

Guide rapide de démarrage ici:

<http://www.mardaso.com/Products/FIRsensorFree/files/ir-blue-android-quick-start.pdf>

La doc du senseur :

<http://melexis.com/Infrared-Thermometer-Sensors/Infrared-Thermometer-Sensors/FIRray16X4-Far-InfraRed-Array-776.aspx>

Explication rapide.

J'ai fait le choix d'utiliser un arduino pro mini 8MHz pour l'adaptation de l'étude originale. La raison principale étant que je voulais que le montage fonctionne directement à partir d'un élément de batterie Li-Ion.

La plus gros problème lié au choix d'un processeur à 8MHz est l'impossibilité de recevoir correctement des chaînes de caractères à 115200 bauds. La seule chaîne que le montage doit recevoir a lieu à l'initialisation. Pour résoudre ce problème la partie a été prise de ne pas interpréter la chaîne de démarrage et de considérer n'importe quelle chaîne comme étant la demande d'initialisation. A priori, cela fonctionne correctement (tant que le protocole n'évolue pas...).

Le senseur est calibré pour une tension inférieure à 3V. On utilise donc une zener 2,7V pour alimenter celui-ci.

On peut voir sur les photos que l'alimentation utilise un petit chargeur de secours que j'ai démonté pour récupérer la batterie et la partie chargeur.

Sur la dernière image, le capteur est placé de telle sorte que lorsque le boîtier est placé devant le téléphone il soit orienté correctement pour que l'image infrarouge se superpose à l'image de la caméra avec la bonne orientation. Le détrompeur situé sur le boîtier (non visible sur la photo) se trouve sur la gauche.

A faire : mesurer la tension de la batterie pour mettre en veille le processeur lorsque la tension est trop faible.

IR-Blue-duino

IR-blue-duino uses Melexis MLX90620 FIR sensor to turn your phone into a thermal cam

inspired by open-source ir_blue project :

<http://www.rhworkshop.com/pages/open-source-ir-blue>

original files :

<https://github.com/RHWorkshop>

Arduino port of basic software found here :

<https://github.com/RHWorkshop/IR-Blue-Eagle-Files/tree/master/IR-Blue%20BT2/Firmware>

The original design was using a PIC micro

IR-blue works with an application on your phone (either Android ones or iPhone)

the phone receives the pixel of the MLX and alpha blend them on top of the phone's 'camera picture

the phone's app can be loaded here:

<https://github.com/RHWorkshop>

This software was only tested with Android application

so make no assumption about the fact it would work with iPhone app

Help for the app can be found here:

<http://www.mardaso.com/Products/FIRsensorFree/Firsensor.html>

Quickstart guide here:

<http://www.mardaso.com/Products/FIRsensorFree/files/ir-blue-android-quick-start.pdf>

Sensor documentation :

<http://melexis.com/Infrared-Thermometer-Sensors/Infrared-Thermometer-Sensors/FIRray16X4-Far-InfraRed-Array-776.aspx>

Some explanation

I made the choice of using an arduino pro mini 8MHz as the platform for that design adaptation.

The main reason was that I wanted the hardware to work on only one Li-Ion element.

The main issue caused by that choice was that the software was unable to receive strings at 115200 bauds. The only string that should be received is at setup time. A trade off was found, the string is not interpreted and any character received is supposed to be the synchronisation string. This trade off seems to work as expected (as long as the protocol don't evolve...)

The thermal sensor is calibrated for a power supply less than 3V. So it is powered from a zener diode.

On the following picture one can see that I used a little power bank that I recycle to use the battery and the charging part.

On the last picture, the sensor is oriented so that when the box is held in front of the phone the infrared image will be correctly oriented when superimposed with the phone's camera picture. The top notch of the sensor (which is hidden on the photo) is pointing to the left

Todo : measuring battery level to idle processor if battery voltage is too low.

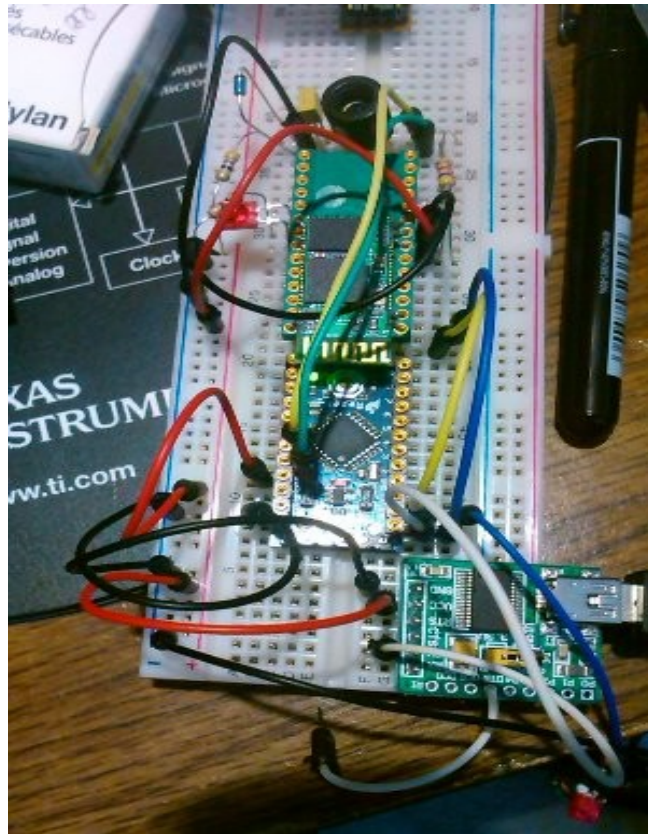


Illustration 1: Proto on breadboard

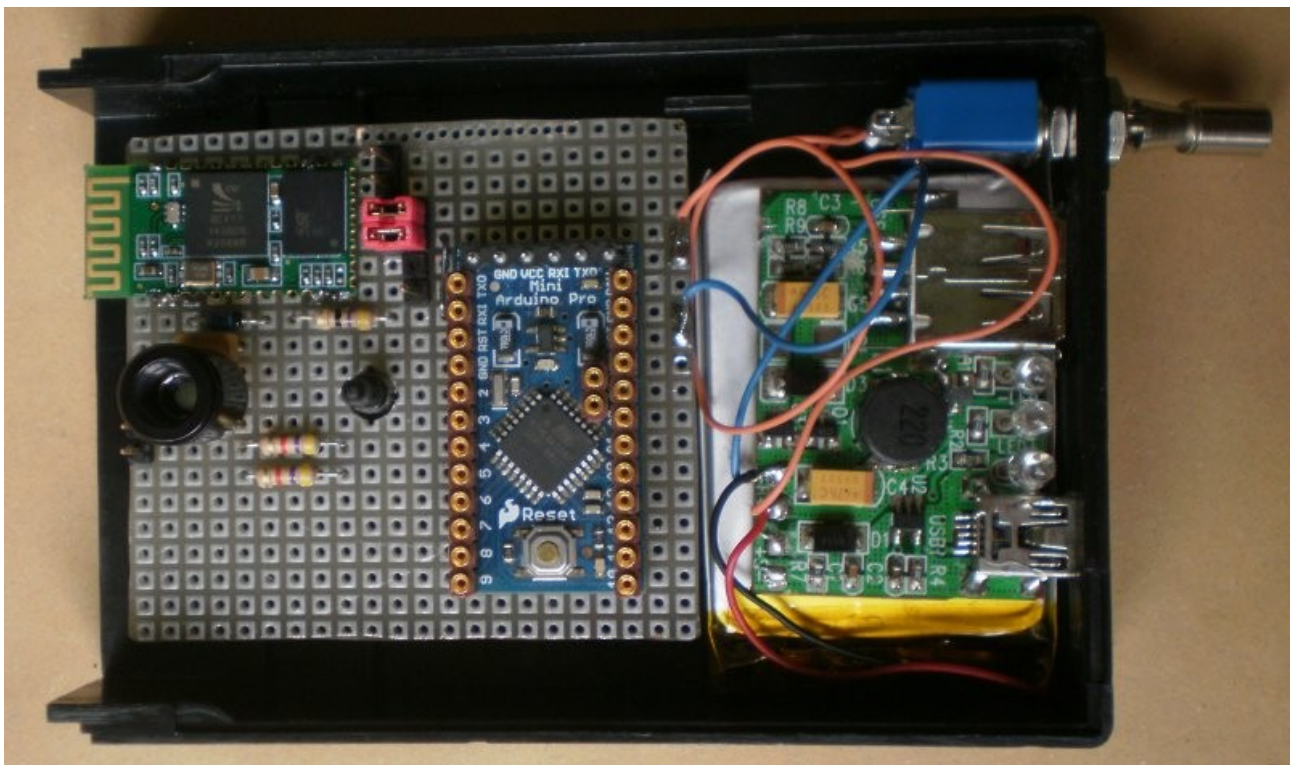
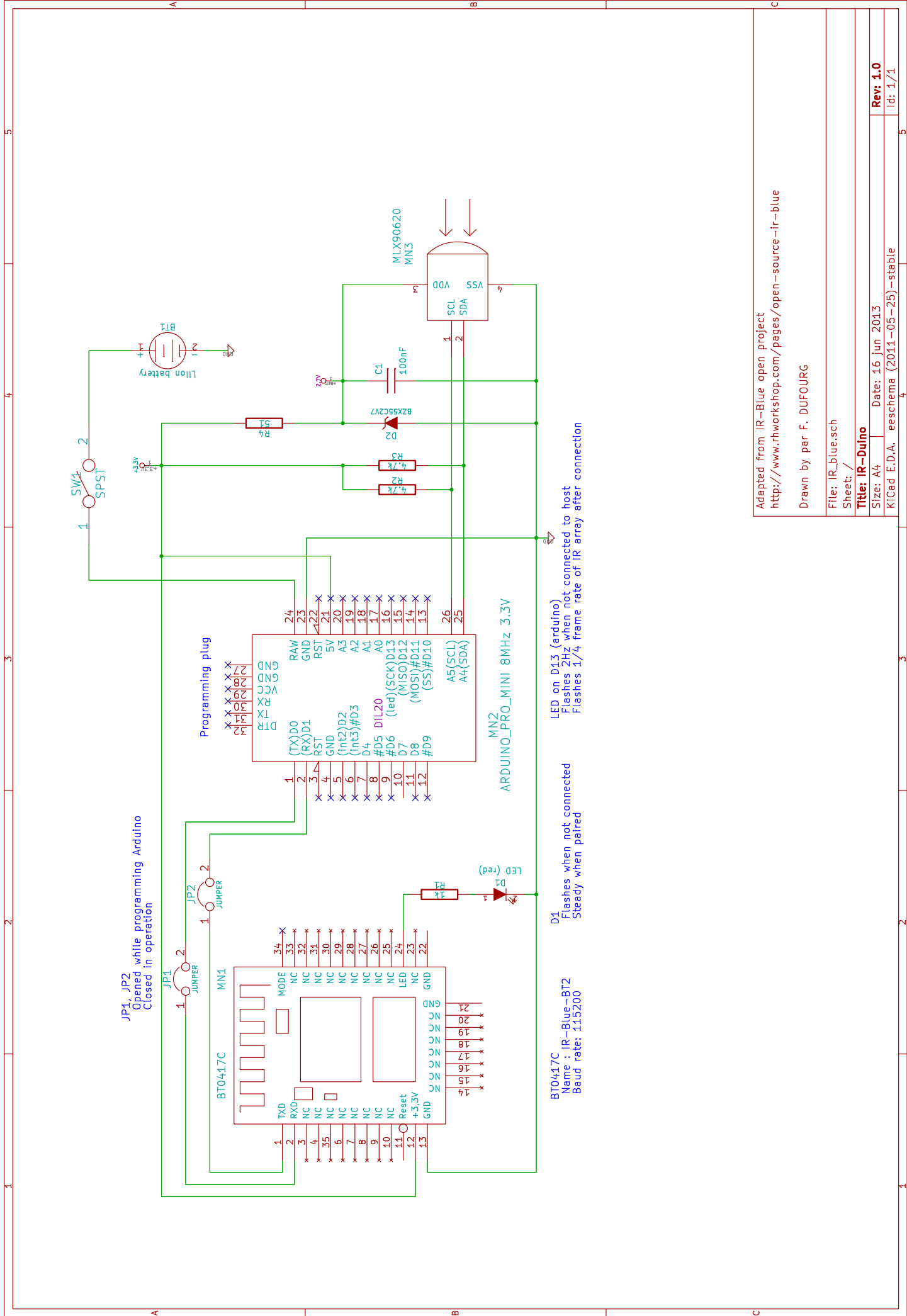


Illustration 2: Integrated proto.

Bill of Material/Nomenclature

ref	value	Estimated Cost
BT1	Li-Ion battery	10,00 €
C1	100nF	0,20 €
D1	LED	0,20 €
D2	BZX55C2V7	0,50 €
JP1	JUMPER	0,25 €
JP2	JUMPER	0,25 €
MN1	BT0417C	8,00 €
MN2	ARDUINO_PRO_MINI 8MHz 3,3V	12,00 €
MN3	MLX90620	70,00 €
R1	1k	0,05 €
R2	4.7k	0,05 €
R3	4.7k	0,05 €
R4	51	0,05 €
SW1	SPST	5,00 €
		106,60 €



Adapted from IR-Blue open project
<http://www.rhworkshop.com/pages/open-source-ir-blue>

Drawn by par F. DUFOURG

File: IR_blue.sch

Sheet: /

Title: IR-Duino

Size: A4 Date: 16 Jun 2013

KiCad E.D.A. eschema (2011-05-25)-stable

Rev: 1.0

Id: 1/1