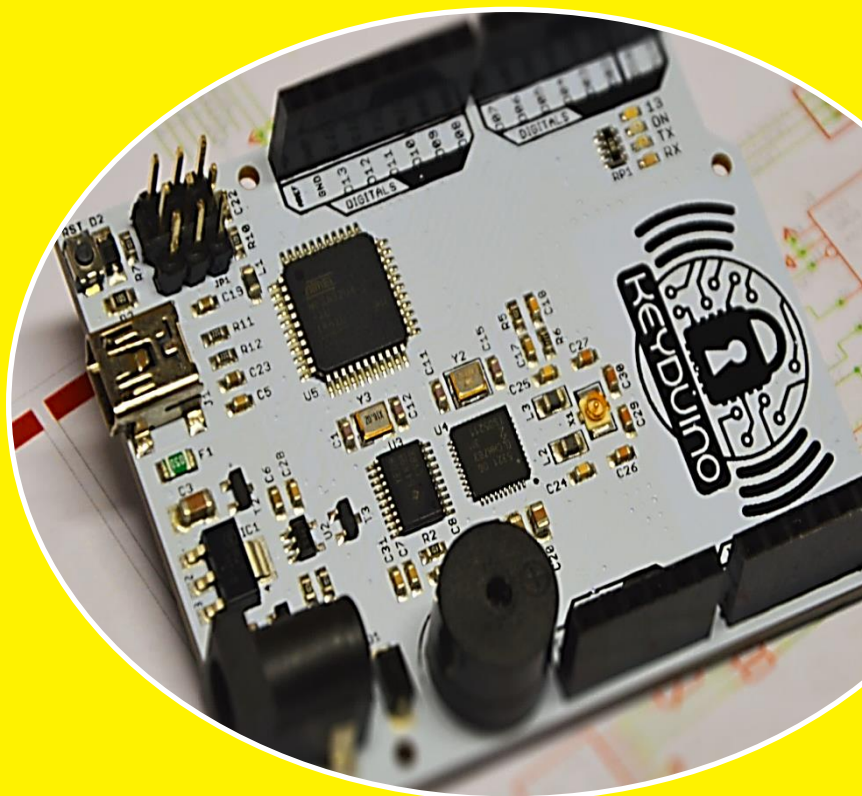


Making Everything Easier!™

KeyDuino

FOR DUMMIES®

KeyDuino	
Which technologies can I use with my KeyDuino ?	
Initialising your KeyDuino	
Getting started with your KeyDuino	
Appendix 1 : circuit diagram	
Appendix 2 : board layout	
Appendix 3 : BOM (component list)	



Introduction

KeyDuino is an NFC (Near Field Communication) development platform which allows the creation of prototype projects quickly and easily. This card is not a specific product for a defined application but is designed to be adapted to create different projects by adding external components and loading software to drive it.

You should pay careful attention to the design and construction of the projects you develop on the KeyDuino platform; the security of the devices you create is entirely your responsibility.

This document provides key information and a description of the techniques you will need to create your own projects.

KeyDuino is a powerful tool and users should respect the ethos that led to its creation - the values of a community of geeks, makers and responsible people aiming to develop and share innovative projects, skills and great ideas.

So please remember, attempting to hack systems that don't belong to you is against the spirit of KeyDuino. A final thought: alone we may be faster, but together we go further!

KeyDuino Team

KeyDuino

KeyDuino is a programmable circuit card based on the Arduino environment (an open source development platform). Keyduino is designed specifically for projects that require interaction with NFC technologies.

KeyDuino is very flexible and can be used to create projects of various kinds in many different fields. From, for example, information transfer utilising NFC, identification of an element on a production line to managing access to a building or vehicle. Its strengths lie primarily in being an efficient solution for prototyping, particularly in the field of security.

Thanks to the numerous inputs and outputs on the card it is easy to establish connections with mechanical devices so it is simple to interact with a door lock, an electronic control system or a car's ignition system.

NFC a key technology these kinds of applications it permits fast and unique identification using a card, NFC ring or Smartphone.

Which technologies can I use with my KeyDuino ?

There are a lot of different tags and devices that are compatible with KeyDuino. Since it is based on an NXP PN532, KeyDuino can communicate with these technologies:

Tags & cards:

- ISO/IEC 14443A/MIFARE Reader/Writer
- FeliCa Reader/Writer
- ISO/IEC 14443B Reader/Writer

Card emulation:

- ISO/IEC 14443A/MIFARE Card MIFARE Classic 1K or MIFARE Classic 4K card emulation mode
- FeliCa Card emulation

Peer-to-peer:

- ISO/IEC 18092, ECMA 340

In terms of NFC, that means you can read NFC Forum Tags Type 1 (Topaz), Type 2 (Mifare Ultralight, or NTAG213 like those that are supplied with KeyDuino), and Type 3 (FeliCa). You can also communicate with NFC Android devices.

If you are completely new to the great big world of contactless technologies, you may be finding this all a little overwhelming!

All you need to know is that these arcane names are just different contactless technologies used in field of NFC security. Each has its own particularities, memory structure, and authentication modes but if you wish to learn more about any of them do some research on the Internet. There's lots information there and even if it seems a bit complicated at first, with a little practice you'll soon have a better understanding of the contactless world. There are some good books on the subject that provide great reference resources if you really want to dive into contactless technologies.

Current practical possibilities

So, we've listed the technologies you can use with your KeyDuino in the wonderful contactless world. That being said, the situation is not always clear and simple - it never is!

ISO/IEC 14443-B tags

The library that we supply to use with KeyDuino is based on diverse developments, starting with the Adafruit PN532 library. At present nobody supplies a support library for ISO/IEC 14443-B tags such as Calypso technology (mostly used in Europe for public transport cards).

This means that while you *can* communicate with a 14443-B card if you send the correct commands with the right settings from your KeyDuino, the library does not at present provide any functions to do that.

If you want to do this you will have to implement the protocol yourself by taking the ISO/IEC 14443-B description and implementing the whole procedure. If you do so, or at least are trying to, please let us know. We would be really glad to add it to our library.

Tags ... tags everywhere

As you may have understood by now, there are a lot of different tags and devices in the contactless and NFC worlds and we cannot test them all. We did a lot of work with Mifare Classic and Mifare Ultralight technologies so these are supported.

Other technologies, like FeliCa, Topaz, and such are not as widespread as the two mentioned above in France, but may be in your country (e.g.: FeliCa is a Sony technology, so it's very often used in Japan). We cannot, for the moment, tell you how to use them with your KeyDuino ; but we would really like to be add support for them in the future.

Once again, if you have feedback about these technologies, we would be really glad to hear from you, so we can share it with the whole community.

NFC Android devices

Guess what ? The same thing goes for Android smartphones!

There are many different Smartphones, each of them using different contactless chipsets. That means that they don't support the same technologies. For example, the old version of Samsung Nexus 7 tablet was great in terms of NFC compatibility. Their newer version however, doesn't support Mifare Classic technology. This is because they changed the NFC chipset in the tablet.

As a consequence you might experience problems when trying peer to peer communication with KeyDuino. It is possible that it won't work with your smartphone. We are really sorry for this and are trying to see what we can do for the future but at present we don't have solutions for every device.

Initialising your KeyDuino

Install the Arduino environment

To start using and programming your KeyDuino, you need to install the Arduino development environment (IDE). You may download versions for PC, Mac OS X and 32/64 bit Linux from here:

<https://www.arduino.cc/en/Main/Software>

Current developments use Arduino version 1.6.5. If you are a new Arduino developer, we advise you to use that same version to avoid problems.

Unzip the file and install the Arduino IDE.

Download KeyDuino library

To get access to the KeyDuino code samples and library, you must download them from GitHub here:

<https://github.com/MrStein/KeyDuino>

Click the « Download ZIP » button on the site and once the download is complete, unzip the contents of the file KeyDuino-master. In the KeyDuino subdirectory you will find several different folders. Copy the one names "KeyDuino" and paste it in your Arduino libraries directory. The library is now installed! You can then start the Arduino IDE and begin to explore.

Getting KeyDuino Android application

We have developed an Android application to experiment exchanges with smartphones. You can find the APK file in the KeyDuino-master/android_app folder.

We will add the source code to the GitHub repository soon. Remember that current Apple iPhones are **not** compatible with the KeyDuino or NFC tags.

Install KeyDuino drivers

Connect KeyDuino board to your computer with the supplied cable. The « On » LED should light.

- **Windows**

For Microsoft Windows :

<http://www.visualmicro.com/post/2012/06/02/Arduino-Leonardo-Windows-Hardware-USB-Installation-Guide.aspx>

- **Linux**

Connect the board. That's all.

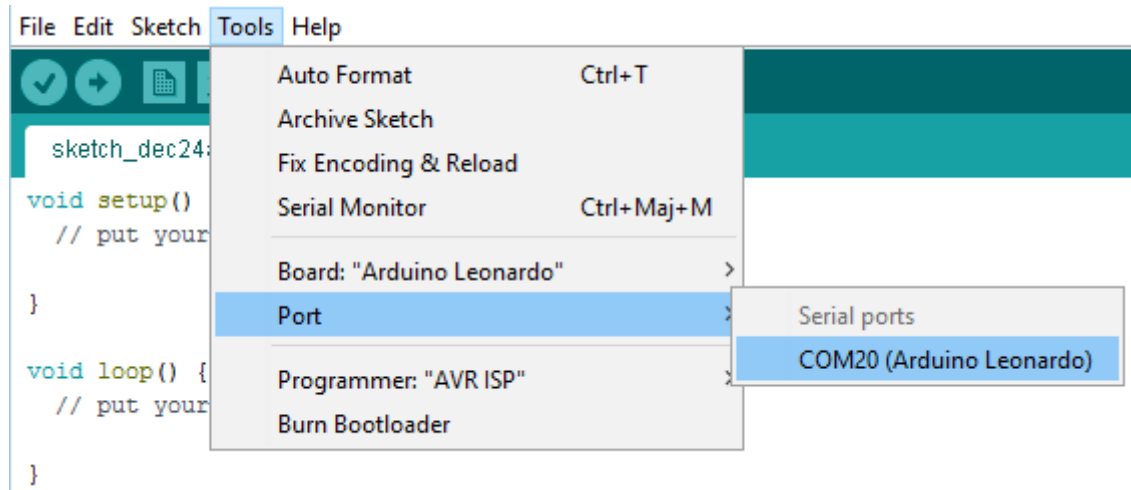
- **Mac OS**

The first time you connect the board on a Mac OS system, the « Keyboard Setup Assistant » should open. Just shut the window, you don't have anything to configure.

Getting started with your KeyDuino

Identify a tag

In the IDE, select the port your board is plugged to. The board should be identified as a Arduino Leonardo.



Error: the port isn't detected

Solution 1: Disconnect your KeyDuino, close the IDE. Reconnect the board and retry selecting the port.

Solution 2 : Re-install the drivers, and check if KeyDuino is recognized by your computer.

Hopefully, the default sketch on the board will allow it to read NFC tags and display the tag's ID.

To see the information generated by the board, click on « Serial Monitor » button.



Error : you're getting the following error message

Board at COM11 is not available

Copy error messages

Check if selected com port is still the same. Sometimes, the port disconnect and change. Then try again. If the problem persists, disconnect the board, shut the monitor, reconnect the board and retry.

For this example, your selected baudrate must be 115200 Baud. By default, if you just downloaded the IDE, it will probably be 9600 ; just change it in the Serial Monitor. Now, take the supplied NFC tag and approach it from KeyDuino antenna ; you should hear a « bip » sound when the board detects it.



Reading antenna



Error: nothing happens

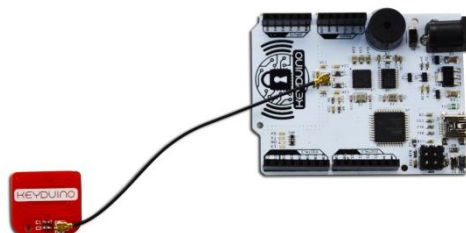
Solution 1: Try with the other tag. If that one does work, please contact us for replacement of the failed tag.

Solution 2: Ensure that the KeyDuino board is well away from any metallic surface during reading.

Solution 3: Try with example PN532_TEST ; if you hear no sound, please contact us. If you hear a sound, try programming the board with tag_identification example which can be found in the [Arduino/libraries/KeyDuino/examples/tag_identification](#) folder.

(Optional) Connecting the antenna

If you ordered a separate antenna, connect it as shown on the schematic below. You should hear a « click ».

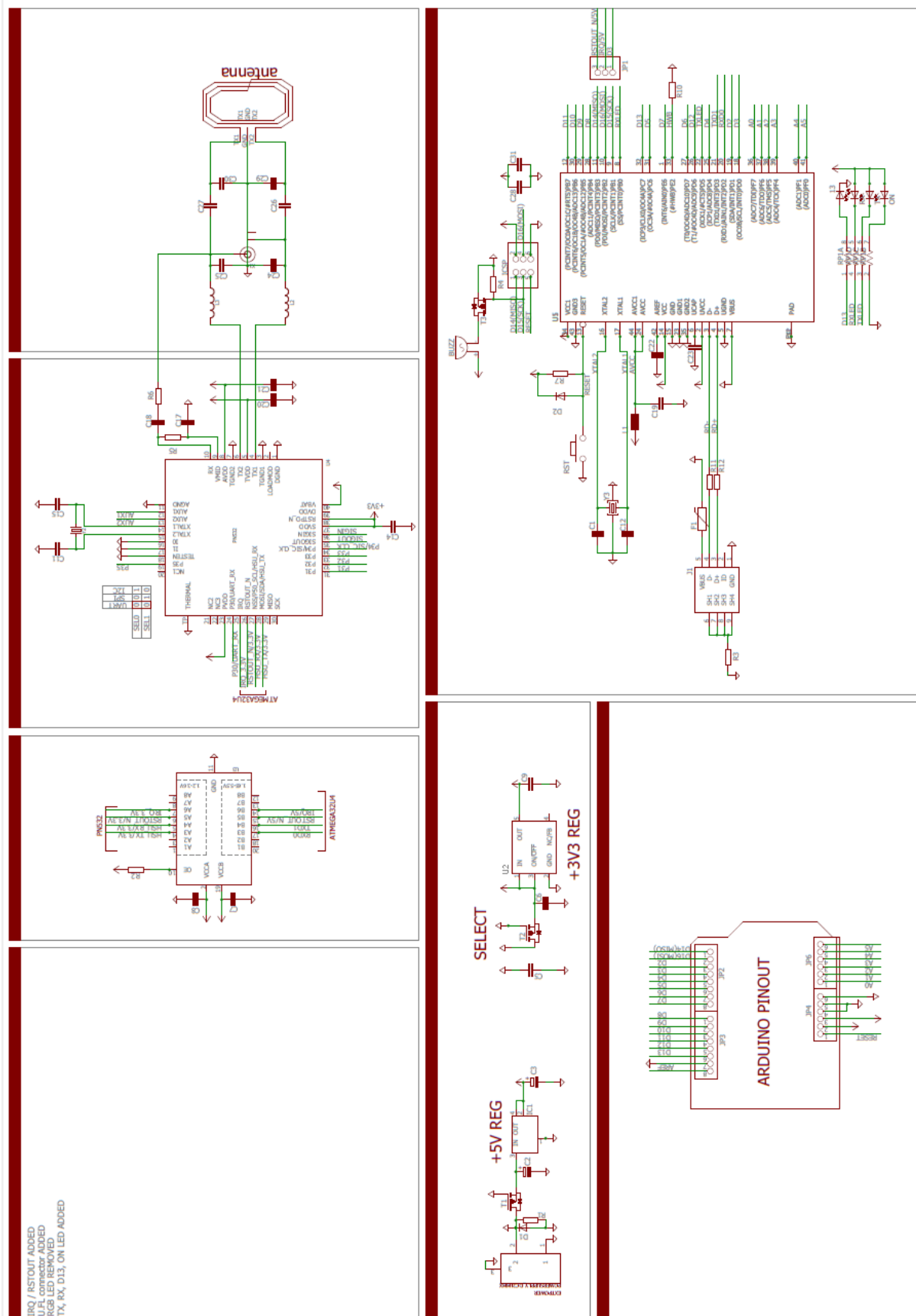


Advice:

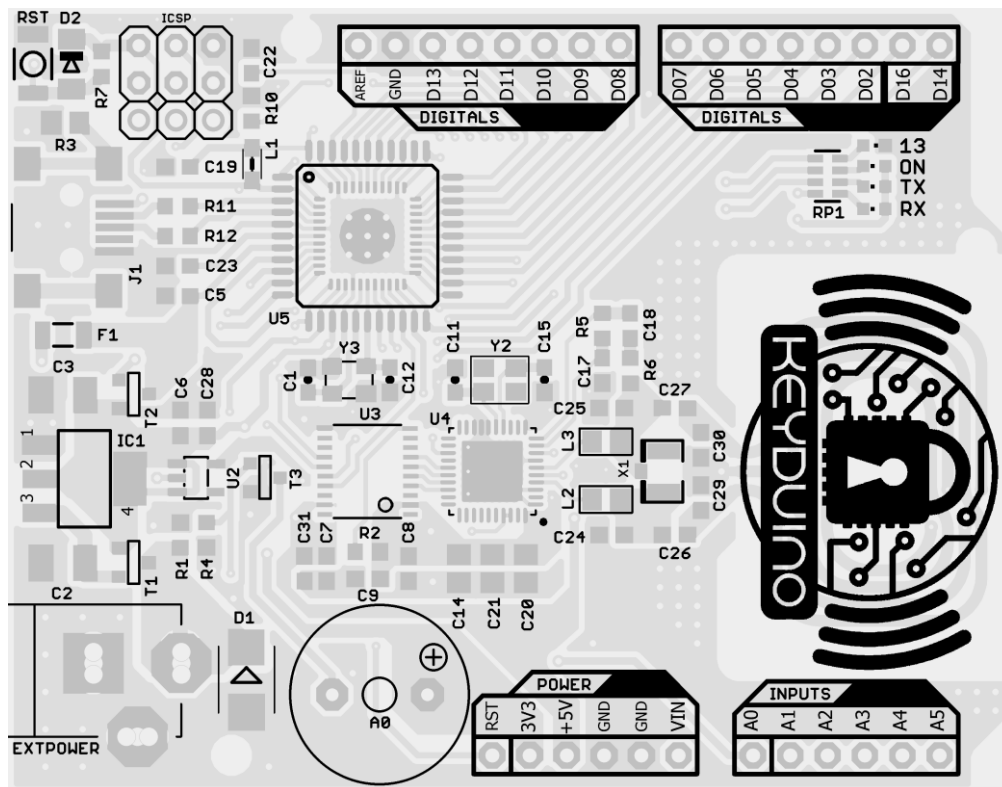
- Try to not bend the cable
- Don't connect and disconnect the cable too often for it is quite fragile. Take care of it.
- Avoid putting it through metallic material (such as a metal case)
- Like integrated antenna, do not allow it to make contact with metal
 - Use plastic screws rather than metallic ones to fix the antenna
 - Note that superglue and double-sided tape work fine!

You are now ready to use your KeyDuino!

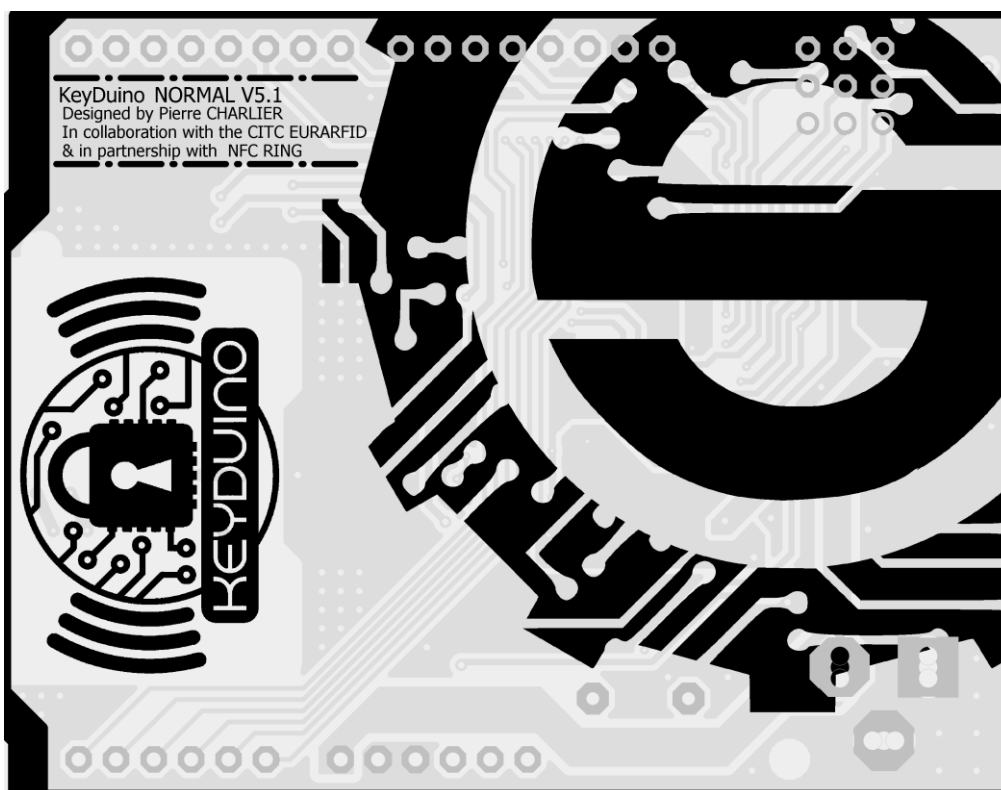
Appendix 1 : circuit diagram



Appendix 2 : board layout



TOP



BOTTOM

Appendix 3 : BOM (component list)

Part	QTY	Value	Device	Package	Description	REF
CAPACITOR						
C1, C11, C12, C15	4	22pF	C-EUC0603	C0603	Capacitor	capacitor 0603 22pF
C24, C25	2	220pF	C-EUC0603	C0603	Capacitor	capacitor 0603 220pF
C26, C27	2	2200pF	C-EUC0603	C0603	Capacitor	capacitor 0603 2200pF
C18	1	1nF	C-EUC0603	C0603	Capacitor	capacitor 0603 1nF
C29, C30	2	1.2nF	C-EUC0603	C0603	Capacitor	Capacitor 0603 1.2nF
C5, C7, C8, C17, C22, C28, C31	7	100nF	C-EUC0603	C0603	Capacitor	capacitor 0603 100nF
C6, C9, C19, C23	4	1uF	C-EUC0603	C0603	Capacitor	capacitor 0603 1uF
C21	1	100nF	C-EUC0805	C0805	Capacitor	capacitor 0805 100nF
C14, C20	2	10uF	C-EUC0805	C0805	Capacitor	capacitor 0805 10uF
C2	1	10uF	CPOL-EUSMCB	SMC_B 1206	CAPACITOR	SMC_B 1206 10uF
C3	1	22uF	CPOL-EUSMCB	SMC_B 1206	CAPACITOR	SMC_B 1206 22uF
RESISTOR						
R11, R12	2		22 R-EU_R0603	R0603	RESISTOR	RESISTOR 0603 22
R5	1	1.0K	R-EU_R0603	R0603	RESISTOR	RESISTOR 0603 1K
R6	1	2.7K	R-EU_R0603	R0603	RESISTOR	RESISTOR 0603 2.7K
R1, R2, R4, R7, R10	5	10K	R-EU_R0603	R0603	RESISTOR	RESISTOR 0603 10K
R3	1	1M	R-EU_R0805	R0805	RESISTOR	RESISTOR 0805 1M
RP1	1		330 RES4NT	RES4NT	4 Resistor Array	4 Resistor Array 330
INDUCTOR						
L1	1	MH2029-300Y	WE-CBF_0805		805 SMD EMI Ferrite	MH2029-300Y
L2, L3	2	560nH	INDUCTOR0805	0805 @ 1	Inductors	inductor 560nH 0805
DIODE & FUSE						
D1	1	M7	DIODE-SMB	SMB	DIODE	SMB M7 (leonardo)
D2	1	CD1206-S01575	DIODE-MINIMELF	MINIMELF	DIODE	CD1206-S01575
F1	1	500mA	PTCSMD	PTC-1206	Resettable Fuse PTC	FUSE 0805 500mA
CONNECTOR & MECHANICAL						
ICSP	1	ICSP	PINHD-2X3	2X03	PIN HEADER	
JP1	1		PINHD-1X3	1X03	PIN HEADER	
JP2	1		PINHD-1X8CLEANBIG	1X08-CLEANBIG	PIN HEADER	
JP3	1		PINHD-1X8CLEANBIG	1X08-CLEANBIG	PIN HEADER	
JP4	1		PINHD-1X6CB	1X06-CLEANBIG	PIN HEADER	
JP6	1		PINHD-1X6CB	1X06-CLEANBIG	PIN HEADER	
X1	1		ANTENNA_U.FL	U.FL	U.FL Antenna Connector	u.FL connector
J1	1	USB-MINI-B% C	USB-MINI-B% C	USB-MINI-B_2		
EXTPOWER	1	POWERSUPPLY_DC21MMX	POWERSUPPLY_DC21MMX	DC-21MM		
RST	1	RESET	PB157	157SW	SKRKAEE010	157SW
BUZZ	1	F/QMX	F/QMX	F/QMX		F/QMX (3.3v)
LEDs						
RX	1		LEDCHIPLED_0603	CHIPLED_0603	LED	0603 LED yellow
TX	1		LEDCHIPLED_0603	CHIPLED_0603	LED	0603 LED RED
ON	1		LEDCHIPLED_0603	CHIPLED_0603	LED	603 LED green
	13	1	LEDCHIPLED_0603	CHIPLED_0603	LED	603 LED blue
IC						
U1	1	ATMEGA32U4-XUAU	ATMEGA32U4-XUAU	TQFP44-PAD	ATMEGA32U4-XUAU	ATMEGA32U4-XUAU
U2	1	LP2985-33DBVR	LP2985-XXDBVR33	SOT23-DBV	ULTRALOW-POWER REGULATORS	LP2985-33DBVR
IC1	1	NCP1117ST50T3G	MC33269ST-3.3T3	SOT223	Regulator 800 mA	NCP1117ST50T3G
U3	1	TXB0108PWR	TXB0108PWR	TSSOP20	8-Bit Bi-Directional Level Shifter	TXB0108PWR
U4	1	PN532	PN532	HVQFN40-6X6	PN532 – NFC controller	PN532
CRYSTAL						
Y2	1	27.12MHz	CRYSTALTHIN	CRYSTAL_3.2X2.5	Crystals	7B-27.1200MAAJ-T
Y3	1	16MHz KX-7	CRYSTAL-3.2-2.5	CRYSTAL-3.2-2.5		16MHz KX-7
MOSFET						
T1	1	PMV48XP	PMOSSOT23	SOT-23	MOS FET	PMV48XP
T2	1	FDN340P	PMOSSOT23	SOT-23	MOS FET	FDN340P
T3	1	BSS123	PMOSSOT23	SOT-23	MOS FET	BSS123