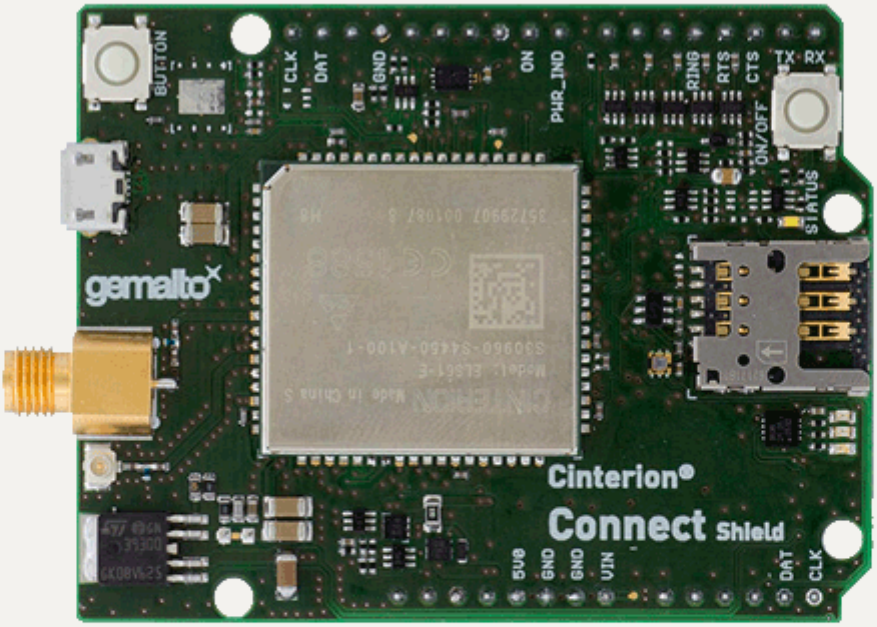
Cinterion® Connect Shield

User Guide

Version: 01
DocId: connect_shield_ug_v01



User Guide:	**Cinterion® Connect Shield**
Version:	**01**
Date:	**2017-01-13**
DocId:	**connect_shield_ug_v01**
Status	**Confidential / Released**

Contents

0 Document History

New document: "Cinterion® Connect Shield User Guide" v01

Chapter	What is new
---	Initial document setup.

1 Introduction

The Cinterion® Connect Shield is designed as a platform for the Internet of Things (IoT).

Programmers looking for wireless connectivity for their sensors and other actors can mount the Connect Shield to their existing Arduino stacks. No hardware expertise is required in most of these cases.

This User Guide describes the Connect Shield's interface capabilities and specifications.

1.1 Package Content

Depending on its configuration, your Connect Shield package includes:
- Cinterion® Connect Shield
- Wideband PCB antenna 700MHz…2.8GHz
- RF cable with U.FL terminations
- SMA to U.FL adaptor
- Quick start guide

1.2 Feature Overview

The Connect Shield provides basic features required for wireless communication
- Technology depending on the ordered Connect Shield configuration: 2G/3G/4G/NB-IoT
- On board I^2C temperature and ambient light sensor as well as 4-bit I^2C extender with 3 LED's connected
- On board micro-SIM card holder
- ON/OFF button plus additional programmable arbitrary button
- Power supply for the Connect Shield through Arduino 5V interface and/or an additional micro-USB on board allowing flexible powering distribution

1.3 Safety Precaution Notes

The common safety precautions that apply to mobile phones must also be observed at all times when using this Connect Shield. Failure to comply with these precautions violates safety standards. Gemalto M2M assumes no liability for customer's failure to comply with these precautions.

The following is a non-extensive list of the mobile phone and Connect Shield usage restrictions:

	Pacemaker patients are advised to keep their hand-held mobile away from the pacemaker while it is on.
	Mobile phones must be switched off before boarding an aircraft.
	Mobile phones may not be operated in the presence of flammable gases or fumes
	Interference can occur if mobile phones are used close to TV sets, radios, computers or inadequately shielded equipment
	Do not use your mobile while driving a vehicle
	You should never rely solely upon any wireless device for essential communications, for example for emergency calls

The power supply connected to the Connect Shield shall be in compliance with the SELV requirements defined in EN 60950-1.

1.4 Regulatory Compliance Information

The Cinterion® Connect Shield is intended for evaluation and development purposes only, and should therefore only be used in a (laboratory) test environment. The device is not CE approved, and has not been authorized as required by the rules of the FCC. All persons handling the Cinterion® Connect Shield must be properly trained in electronics and observe good engineering practice standards.

2 Interfaces

2.1 Overview

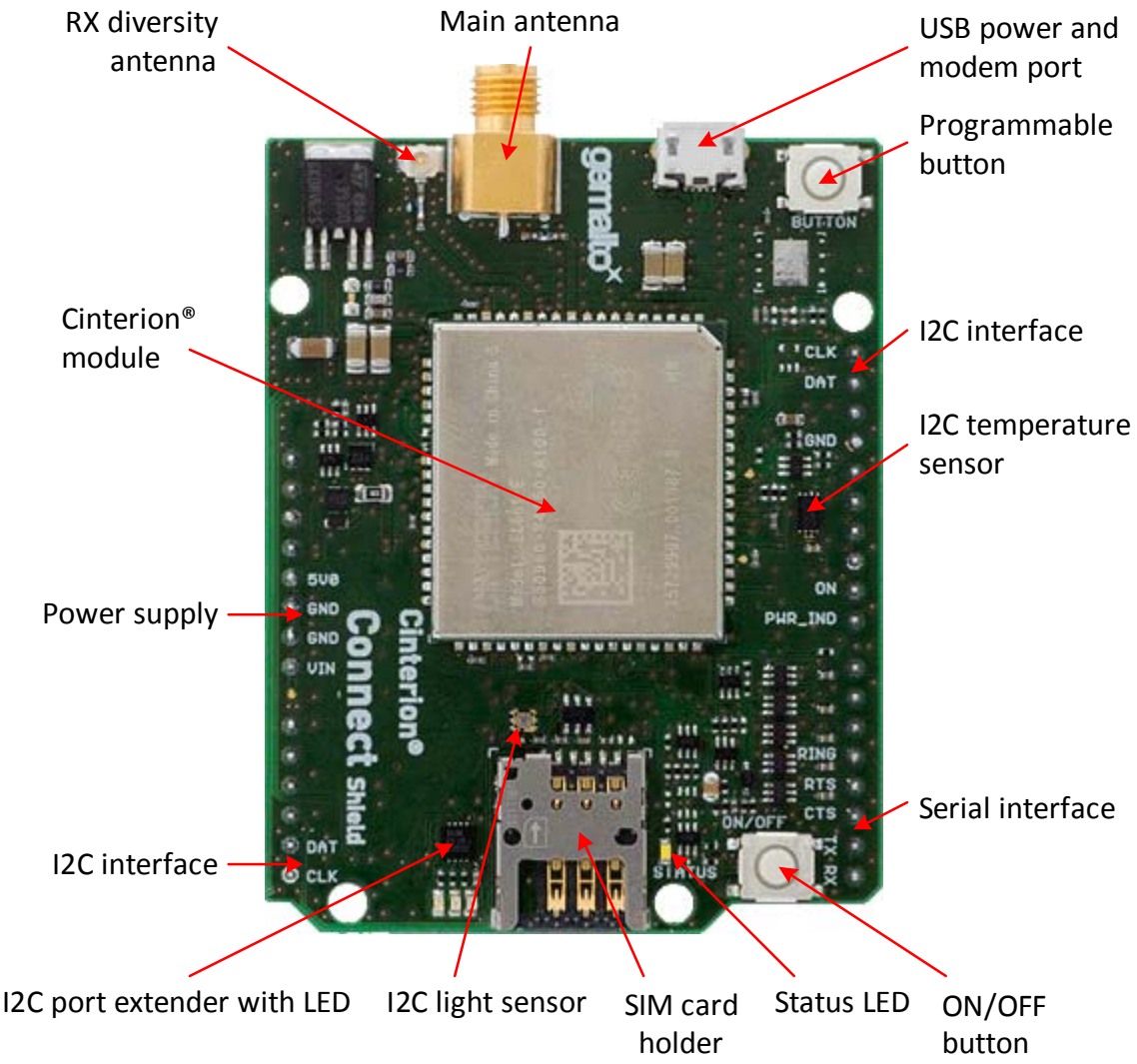


Figure 1: Connect Shield interfaces

2.2 Application / Arduino UNO Interface

The Cinterion® Connect Shield uses a few Arduino ONE interface signals, and it is possible to combine the Connect Shield with other Arduino shields. To occupy as few signals as possible, the Connect Shield output lines PWR_IND, RING, and CTS are connected to the Arduino interface via output resistors, thus enabling these signals for a further usage, for example between the Arduino and another sensor shield. In this case the Arduino's or sensor shield's pin drive strength must be higher than 5mA to overrule the Connect Shield's output.

2.2.1 Pin Outline

A detailed pin description of the Arduino Uno interface is given following two tables.

Shield I/F	Shield function	Shield DIR	Shield Interface (Concept Shield)	Shield DIR	Shield function	Shield I/F
				I	SCL	SCL
				IO	SDA	SDA
						AREF
				GND	GND	GND
NC	-	-		-	-	SCK
IOREF	-	-		-	-	MISO
RESET	-	-		-	-	MOSI
3.3V	PWR	I		-	-	CS
5V	PWR	I		I	ON/OFF	D9
GND	PWR	GND		O	PWR_IND	D8
GND	PWR	GND		-	-	D7
Vin	-	-		-	-	D6
A0	-	-		-	-	D5
A1	-	-		O	RING	D4
A2	-	-		I	RTS	D3
A3	-	-		O	CTS	D2
SDA	SDA	IO		I	TXD	TX
SCL	SCL	I		O	RXD	RX

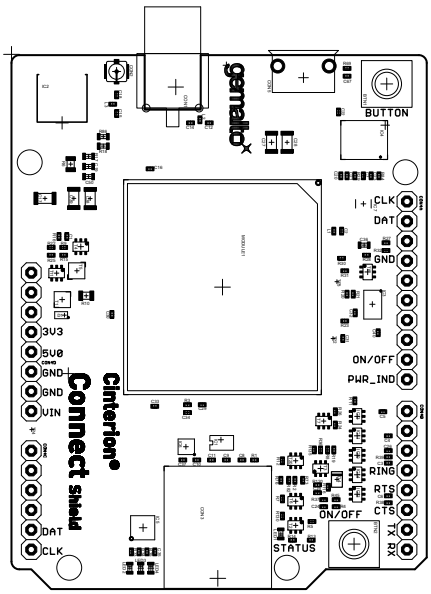


Figure 2: Connect Shield's Arduino UNO interface lines

Table 1: Connect Shield's Arduino UNO interface pin and signal description

Name	Modem Reference	Dir	Description	Properties
NC	-		Not connected	
IOREF	-	-	Not connected	
RESET	-	-	Not connected	
3.3V	-	I	Shield supply	I^2C sensor supply
5V	-	I	Shield supply	Modem supply via voltage regulator
GND	-		Main Ground	
Vin	-	-	Not connected	
A0	-	-	Not connected	
A1	-	-	Not connected	
A2	-	-	Not connected	
A3	-	-	Not connected	
SDA	SDA	IO	I2C data signal	5V compliant, Rpu=4.7kΩ, connected to modem via level shifter
SCL	SCL	O	I2C clock signal	5V compliant, Rpu=4.7kΩ, connected to modem via level shifter
SCL	SCL	O	I2C clock signal	5V compliant, Rpu=4.7kΩ, connected to modem via level shifter
SDA	SDA	IO	I2C data signal	5V compliant, Rpu=4.7kΩ, connected to modem via level shifter
AREF	-	-	Not connected	
SCK	-	-	Not connected	
MISO	-	-	Not connected	
MOSI	-	-	Not connected	
CS	-	-	Not connected	
D9	ON	I	ON/OFF	A falling edge triggers the modem ON signal. A low level (>3s) switches off the modem power supply. Note: After the falling edge this signal should be released within 1s in order to keep the modem running.
D8	PWR_IND	O	PWR_IND	Low level open collector output, Rout=100Ω, indicating the modem has been started
D7	-	-	Not connected	
D6	-	-	Not connected	
D5	-	-	Not connected	
D4	RING	O	Host wake up	5V compliant, Rout=1kΩ, wake host from sleep for incoming message
D3	RTS	I	Flow control	5V compliant, Iinmax=+-2µA, Modem wake up, can be configured as not used
D2	CTS	O	Flow control	5V compliant, Rout=1kΩ, can be configured as not used
TX	RxD0	O	Serial out	5V compliant, Push pull; Io=+-50mA
RX	TxD0	I	Serial in	5V compliant, Iinmax=+-2µA

2.2.2 UART

The Connect Shield interface supports a UART (RxD/TxD) connection with up to 921kbit/s. By default the speed is set to 115kbit/s. It is mandatory to use the modem's flow control RTS in case the modem is configured for SLEEP mode (i.e., power saving). In SLEEP mode - please refer to the modem's AT Command Set for possible SLEEP mode settings - the modem awakes by a falling RTS signal and retrieves UART data. On incoming messages the RING toggles to awake the host from sleep, this line should be interrupt able on host side.

2.2.3 USB

The Connect Shield comes with a separated micro-USB interface that is intended to

- Supply additional power in case of weak Arduino supply while transmitting data in areas with a weak network coverage
- Support additional serial interfaces for different purpose, like debugging, modem firmware upgrade or command interface

Whenever the modem switches off while transmitting data over the air, an additional USB power supply 5V/1A should be connected to resolves such power issues. Please note, this additional supply is not fed toward the Arduino base board, meaning the Arduino has to be supplied in any case.

Note: The micro-USB driver can be downloaded under developer.gemalto.com.

2.2.4 I^2C

Arduino's I^2C interface is connected to the modem as well as to a separate temperature sensor, ambient light sensor and a 4-bit port extender.

Note: The modem can be master only. For I^2C commands and settings for modem controlled I^2C please refer to the modem's AT Command Set. Refer to the device command description on how to setup the I^2C protocol.

2.2.4.1 I^2C Temperature Sensor

On board is the temperature sensor NXP PCT2075 accessible via I2C address 1110 101 (0x75).

2.2.4.2 I^2C Ambient Light Sensor

On board is the ambient light sensor Vishay VEML6030 accessible via I2C address 0010 000 (0x10).

Note: Ambient light sensor needs to be powered by the 3.3V source from Arduino based board.

2.2.4.3 I²CPort Extender

On board is 4-bit port extender PCA9536TK accessible via I²C address 1000 001 (0x41). Three IO are connected to LED's as output, while one IO is free for any usage on a test point.

Table 2: I²C port extender

Hex	TP(IO3)	LED Yellow(IO2)	LED Orange(IO1)	LED Green(IO0)
0x00	low	on	on	on
0x01	low	on	on	off
0x02	low	on	off	on
0x03	low	on	off	off
0x04	low	off	on	on
0x05	low	off	on	off
0x06	low	off	off	on
0x07	low	off	off	off
0x08	high	on	on	on
0x09	high	on	on	off
0x0A	high	on	off	on
0x0B	high	on	off	off
0x0C	high	off	on	on
0x0D	high	off	on	off
0x0E	high	off	off	on
0x0F	high	off	off	off

2.2.4.4 I²C Quick Access via Module I²C

Please complete the following steps for a quick access via the module's I²C lines:

- Download the installation package from developer.gemalto.com.
- Connect the USB cable from the PC to the Connect Shield's USB socket
- Start your Connect Shield with the ON button (short press).
- Install the modem USB drivers from the ZIP file. The drivers will have to be installed manually.
- Open the Windows Device Manager to check for the modem's COM Port number: Modems > Cinterion <Modem name> Java Debug Modem USB > Properties > Modem > Port
- Open your favorite Terminal program, and connect to Connect Shield using these settings: < COM port number>, 115200 bps, 8N1, CTS/RTS control, DTR on
- Using the Terminal program you can now start entering commands for the Connect Shield (plus CR+LF) as shown further below.

The given addresses for the sensors and the port extender are 7 bit. Adding after the address a read bit R=1 or write bit W=0 changes the length to 8bit.

PCT2075 temperature sensor	PCA9536TK port extender	VEML6030 ambient light sensor
0x75 + W = 0xEA	0x41 + W = 0x82	0x10 + W = 0x20
0x75 + R = 0xEB	0x41 + R = 0x83	0x10 + R = 0x21

Example for reading the PCT2075 temperature sensor:

To read the temperature via module I^2C you need to open the I^2C bus:

AT^SSPI=	(Wait for "*CONNECT*") (EA=0x75+W, 00=temperature pointer byte)
<aEA00>	(Answer: *{a+}*) (EB=0x75+R, 0002=16bit which needs to be parsed)
<aEB0002>	(Answer: *{a+1A20}*) (Close the I^2C bus)
#	(Answer: *"OK"*)

The returned answer is for example *1A20* $_{hex}$ => $0001.1010.0010.0000_{bit}$
The first 7 MSB should be used => 0001.1010.0010.0000
$0001.1010.001_{bit}$ = 209 $_{dec}$
209 x 0,125°C = 26°C

Example for reading the PCA9536 port extender:

To set the port extender via module I^2C you need to open the I^2C bus:

AT^SSPI=	(Wait for "*CONNECT*") (82=0x41+W, 0x03=Configuration Register, 0xF0=sets Q0 to Q3 to output)
<a8203F0>	(Answer: *{a+}*) (82=0x41+W, 0x01=Output Port Register, 0xF0= turns all 3 LEDs on and test point to low)
<a8201F0>	(Answer: *{a+}*) (Close the I2C bus)
#	(Answer: *"OK"*)

2.3 Buttons

Pressing the ON/OFF button shortly starts the modem - indicated by the LED lighting up.

Pressing the ON/OFF button for more than 3s switches off the modem's power supply, i.e., powers off the modem, which is indicated by the LED going off.

Note: An Arduino may come with a default low level at the ON signal that causes the Connect Shield to have the power supply permanently disabled. Such a case is indicated by the LED remaining off when stacked up.

A further button is freely available, and might be polled by the modem or Arduino.

Note: The Connect Shield hardware revision A2 does not support the button being controlled by the Arduino.

3 RF Interference

When stacking up the Connect Shield with other shields as well the Arduino together with the PCB antenna, high power RF radiation from the high efficiency PCB antenna may interfere with other shields or the Arduino, especially in 2G mode with up to +33dBm output power. This interference may results in signal distortion caused by RF demodulation. In such cases it is recommended to separate the antenna from the shield and increase the distance.

4 Characteristics

4.1 Limits

Table 3: Absolute maximum ratings

Parameter	Min	Max	Unit
Supply voltage on USB ports	-0.3	5.5	V
Voltage at 3.3V application supply interface (Arduino)	-0.3	3.6	V
Voltage at 5V application supply interface (Arduino)	-0.3	5.25	V
Voltage at application signal interface (Arduino)	-0.3	6	V
Environmental temperature	0	35	°C

Note: Violation of these limits may cause permanent damages to the Connect Shield.

4.2 Recommended Operating Conditions

Table 4: Recommended operating conditions

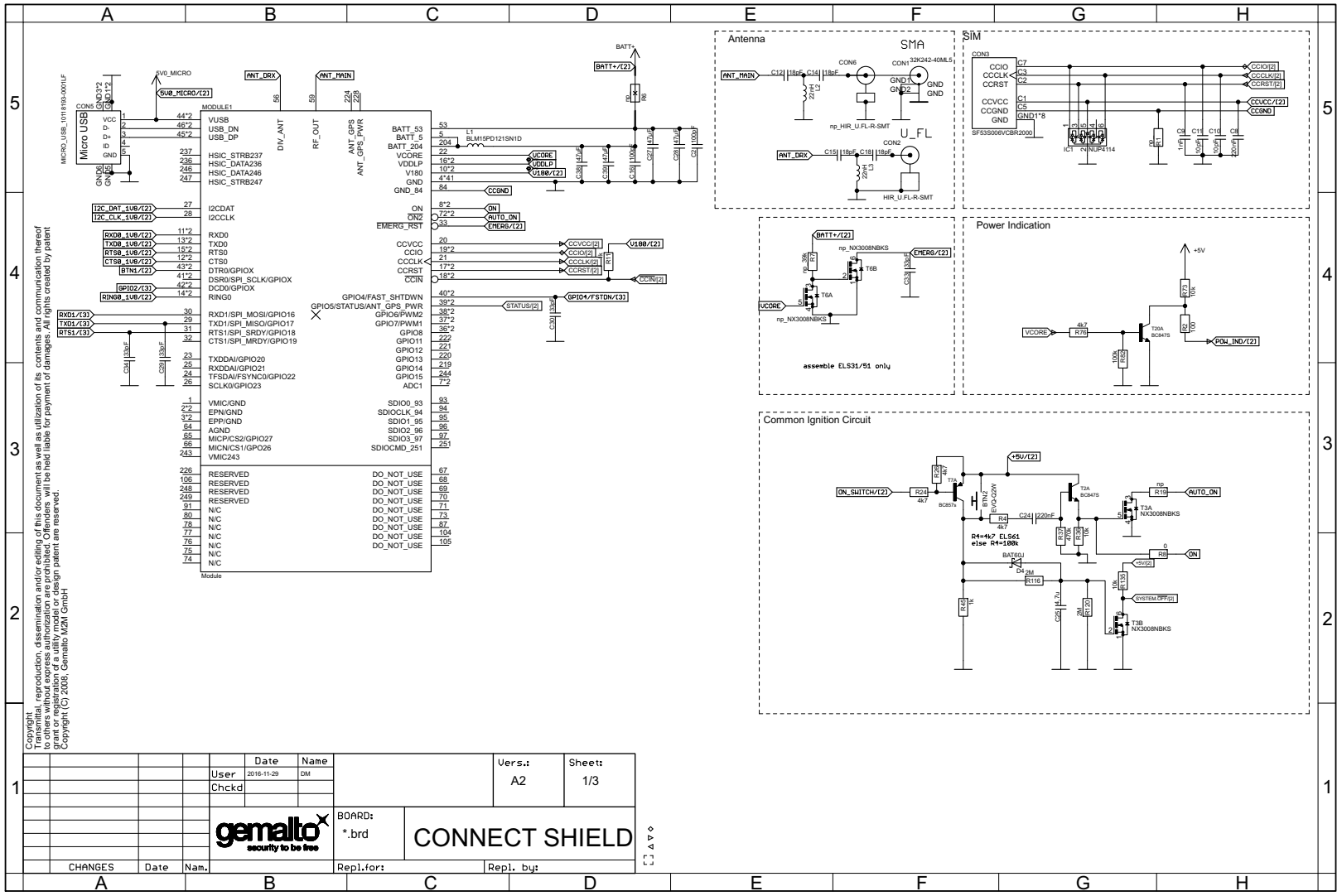
Parameter	Min	Max	Unit
Supply voltage on USB ports	4.75	5.25	V

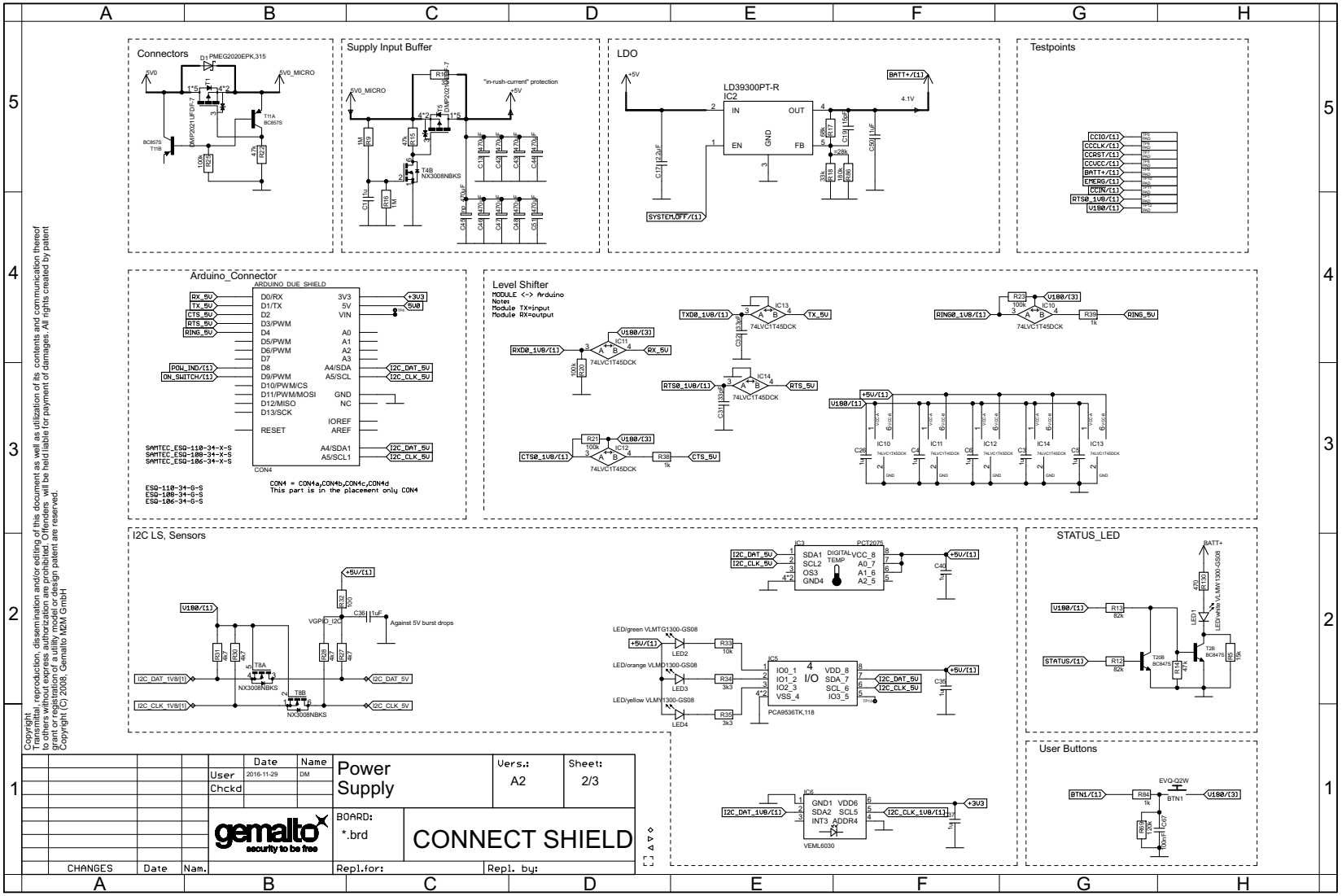
4.3 Static Characteristics

Table 5: Static characteristics

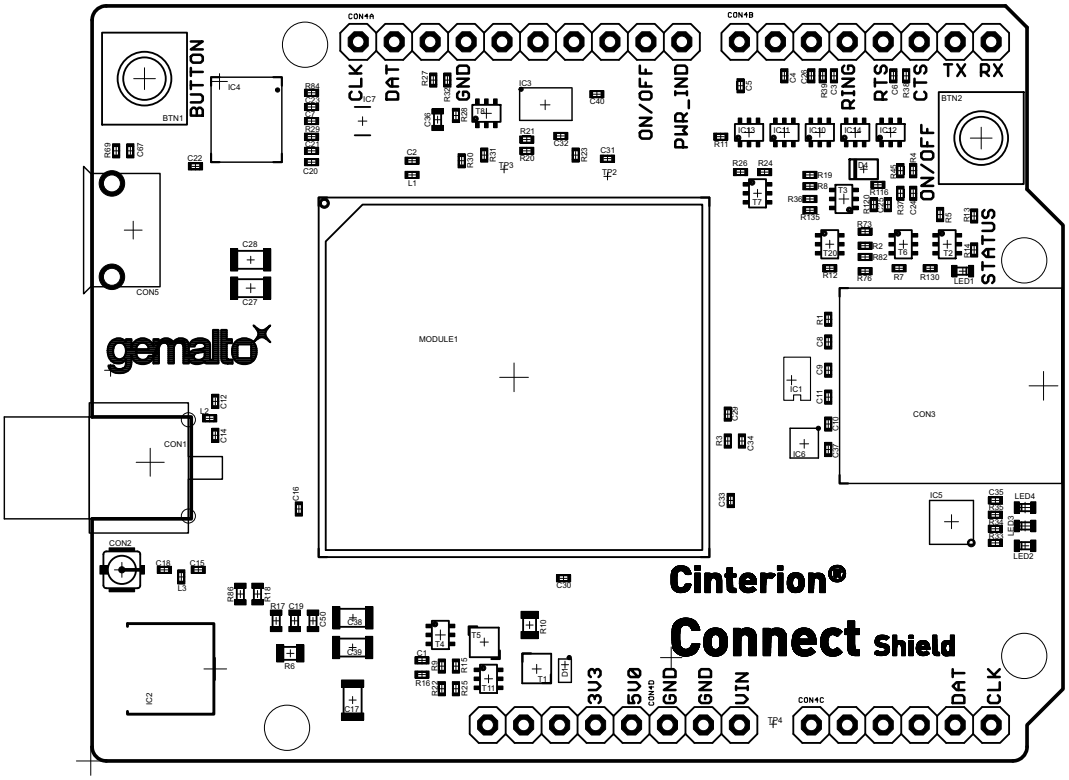
Parameter	Max value	Unit
High level input voltage on application interface	3.36	V
Low level input voltage on application interface	1.44	V
High level output voltage on application interface @ Vusb=5V; Io=-32mA	TBD.	V
Low level output voltage on application interface @ Vusb=5V; Io=32mA	TBD.	V

5 Appendix – Schematics




Connectors
Supply Input Buffer
LDO
Testpoints
Arduino_Connector
Level Shifter
I2C LS, Sensors
STATUS_LED
User Buttons
LD39300PT-R
IC2
ARDUINO_DUE_SHIELD
CON4
SAMTEC_ESQ-110-34-X-S
SAMTEC_ESQ-108-34-X-S
SAMTEC_ESQ-106-34-X-S
ESQ-110-34-G-S
ESQ-108-34-G-S
ESQ-106-34-G-S
CON4 = CON4a,CON4b,CON4c,CON4d
This part is in the placement only CON4
"in-rush-current" protection
74LVC1T45DCK
PCT2075
PCA9536TK,118
VEML6030
LED/green VLMTG1300-GS08
LED/orange VLMO1300-GS08
LED/yellow VLMY1300-GS08
LED/white VLMW1300-GS08
Copyright
Transmittal, reproduction, dissemination and/or editing of this document as well as utilization of its contents and communication thereof to others without express authorization are prohibited. Offenders will be held liable for payment of damages. All rights created by patent grant or registration of a utility model or design patent are reserved.
Copyright (C) 2008, Gemalto M2M GmbH
Power Supply
Vers.: A2
Sheet: 2/3
BOARD: *.brd
CONNECT SHIELD
User 2016-11-29 DM
Chckd
CHANGES Date Nam.
Repl.for:
Repl. by:

5.1 Placement



About Gemalto

Gemalto (Euronext NL0000400653 GTO) is the world leader in digital security with 2015 annual revenues of €3.1 billion and blue-chip customers in over 180 countries. Our 14,000+ employees operate out of 118 offices, 45 personalization and data centers, and 27 research and software development centers located in 49 countries.

We are at the heart of the rapidly evolving digital society. Billions of people worldwide increasingly want the freedom to communicate, travel, shop, bank, entertain and work - anytime, everywhere - in ways that are enjoyable and safe. Gemalto delivers on their expanding needs for personal mobile services, payment security, authenticated cloud access, identity and privacy protection, eHealthcare and eGovernment efficiency, convenient ticketing and dependable machine-to-machine (M2M) applications.

Gemalto develops secure embedded software and secure products which we design and personalize. Our platforms and services manage these secure products, the confidential data they contain and the trusted end-user services they enable. Our inovations enable our clients to offer trusted and convenient digital services to billions of individuals.

Gemalto thrives with the growing number of people using its solutions to interact with the digital and wireless world.

For more information please visit
m2m.gemalto.com, www.facebook.com/gemalto, or Follow@gemaltom2m on twitter.

Gemalto M2M GmbH
Werinherstrasse 81
81541 Munich
Germany

➲ M2M.GEMALTO.COM

gemalto
security to be free