

Manuale AGEduino MINI

di Umberto Allievi

umberto.allievi@agevoluzione.it

Milano, 19/08/15

Indice generale

Introduzione.....	3
La Gamma AGEduino MINI.....	5
Model A.....	5
Model B.....	5
Caratteristiche.....	7
Comparazione con Arduino UNO/MEGA/Pro Mini.....	7
Immagini.....	8
Fronte.....	8
Retro.....	8
Mappa Funzionale.....	9
Fronte.....	9
Retro.....	10
Pinout Scheda.....	11
Pin Mapping processore.....	12
Uscite PWM transistorizzate – LED Driver.....	14
Configurazione.....	15
Alimentazione processore.....	15
Alimentazione scheda WiFi.....	16
Autoreset connessione seriale.....	17
Connessioni.....	18
Alimentazione.....	18
Seriale di Programmazione.....	20
Scheda WiFi.....	21
Connessione ICSP.....	22
Rimozione Scheda Supporto.....	23
Configurazione IDE Arduino.....	24
Prima inizializzazione del processore.....	25
Installazione supporto ICSP AVR-ISP500-TINY.....	29
Installazione Bootloader.....	30
Programmazione tramite ICSP.....	30
Programmazione Tramite USB/Seriale.....	30
Supporto scheda WiFi.....	31
Programmi di Esempio.....	32
Installazione IDE 1.6.3 Sperimentale.....	33
Installazione IDE 1.7.4 Sperimentale.....	34

Introduzione

Il progetto AGEduino è derivato dal progetto Arduino e può utilizzare le librerie e l'IDE di Arduino in quanto il processore utilizzato è supportato dal progetto Arduino grazie alle estensioni Sanguino.

Da questo si **differenzia** per alcune caratteristiche chiave:

- **dimensioni ridotte**

le dimensioni di circa 74-60x34 mm permettono un facile inserimento anche in oggetti di piccole dimensioni. La riduzione della dimensione è ottenuta senza sacrificare l'espandibilità: tutte le porte del processore vengono riportate sui connettori con passo 2,54 mm. La scheda è scomponibile da due parti: la parte con il connettore ICSP è removibile.

- **capacità aumentate**

l'utilizzo del processore ATMEL ATmega1284P a 16 Mhz permette di mantenere piccole dimensioni ma nel contempo di quadruplicare le risorse di memoria disponibili rispetto ad Arduino UNO, grazie ai suoi 128KB di Flash, 16KB di RAM e 4K di EEPROM.

- **due UART (seriali TTL)**

Il processore scelto rende disponibili due connessioni seriali asincrone, rispetto a 1 nell'Arduino UNO.

- **interfaccia WiFi**

AGEduino dispone di un connettore che permette l'utilizzo del controller WiFi CC3000 di Texas Instruments. Grazie alle librerie dedicate la gestione della comunicazione WiFi è facile ed affidabile. Il connettore è stato realizzato per la connessione al CC3000 Breakout di Sparkfun

- **alimentazione switching**

grazie all'alimentazione switching AGEduino è ideale per i progetti che fanno del risparmio energetico il loro obiettivo, e permette di creare dispositivi alimentati da batteria che ne ottimizzano la durata. AGEduino può essere alimentato in corrente continua da 5V fino a 20V.

Inoltre l'alimentazione del microcontrollore è configurabile a 3.3V, 4V e 5V per adattarsi alle esigenze dell'utente.

- **LED Driver**

quattro uscite PWM possono comandare i driver per strisce di luci LED montati sul retro della scheda. Crea il tuo progetto di illuminazione senza aggiungere altri componenti esterni.

- **oscillatore al Quarzo**

per un funzionamento più affidabile e preciso abbiamo installato un oscillatore al quarzo a 16MHz. Questo permette al processore di funzionare in modo affidabile anche a 3.3V, tensione inferiore a quanto specificato da ATMEL

- **assenza del convertitore da seriale TTL a USB, del bottone di reset e di led on-board**

per minimizzare le dimensioni bisogna sacrificare qualcosa: abbiamo tolto tutto ciò che non è essenziale in un progetto embedded. Per programmare il dispositivo si può utilizzare uno dei convertitori in commercio. È stato predisposto un connettore per la programmazione compatibile con il convertitore USB-Seriale di Sparkfun

- **made in italy**

qualità italiana per un prodotto che non è solo per la realizzazione di prototipi, ma aspira a entrare nel mondo embedded industriale.

La scheda AGEduino è divisa in due parti staccabili:

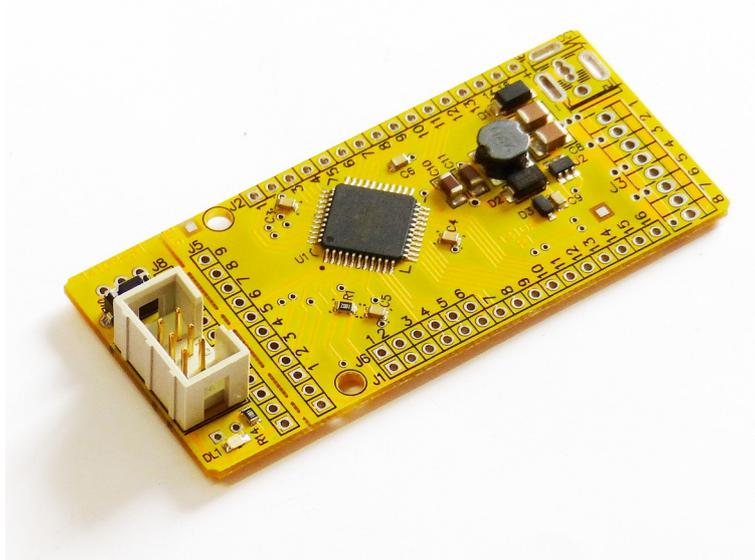
- la parte più piccola, denominata **scheda di supporto**, contiene:
 - LED di alimentazione
 - Connettore a vaschetta ICSP
 - Pulsante di reset
- mentre la parte più grande, denominata **scheda madre**, contiene:
 - Processore
 - Quarzo
 - Circuito di alimentazione
 - MOSFET per alimentazione LED

Questa documentazione è disponibile sul sito www.agevoluzione.com/ageduino , ulteriori domande possono essere postate sul sito wiki

La Gamma AGEduino MINI

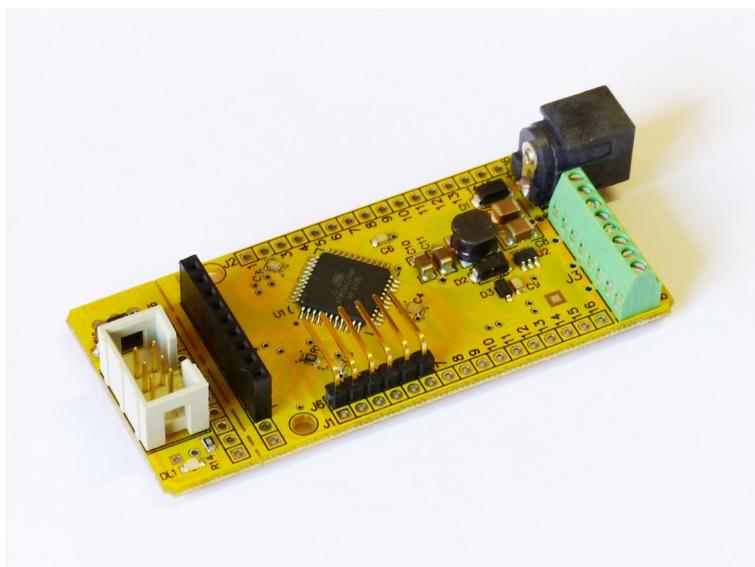
AGEduino MINI viene commercializzato in due versioni:

Model A



Il Model A monta solamente il connettore ICSP sulla scheda di supporto. Tutti i connettori della scheda madre non sono montati. Questo per permettere all'utilizzatore di montare i connettori più adatti alla installazione che si desidera realizzare.

Model B



Il Model B viene fornito completo dei seguenti connettori:

- Strip femmina 9 poli per connessione a scheda WiFi

- Strip maschio 6 poli per connessione adattatore USB-Seriale
- Connettore di alimentazione coassiale 2.1 mm con polo positivo al centro
- Connettore a vite a 8 poli per connettere fino a 4 strisce di LED

Caratteristiche

Comparazione con Arduino UNO/MEGA/Pro Mini

Caratter/Modello	AGEduino	Arduino UNO	Arduino MEGA	Arduino Pro Mini
Processore	ATMega1284	ATMega328	ATMega2560	ATMega328
RAM	16K	2K	8K	2K
FLASH	128K	32K	256K	32K
EEPROM	4K	1K	4K	1K
Alimentazione	Switching	Lineare	Lineare	Lineare
Tensione alimentazione	Da 4.0 a 20 V	Da 7.0 a 12 V	Da 7.0 a 12 V	Da 5.0 a 12 V
Tensione Micro	5.0 , 4.0, 3.3 V	5.0 V	5.0 V	5.0 V
N. Seriali Async.	2	1	2	1
N. PWM	6	6	15	6
N. I/O Pins	32	14	54	14
Analog Pins	8	6	16	8
Clock Speed	16Mhz	16Mhz	16Mhz	16Mhz
Oscillatore	quarzo	risuonatore	risuonatore	risuonatore
Lunghezza	74/60 mm*	68 mm	101 mm	33 mm
Larghezza	34 mm	53 mm	53 mm	18 mm
USB connection	FTDI cable	Si	Si	FTDI cable
WiFi Pinout	Si**	Si	Si	No
MOSFET	4	0	0	0
Supporto Arduino IDE	Si	Si	Si	Si
Reset Button	Si***	Si	Si	Si
Reset Automatico Seriale programm.	Selezionabile	Si fisso	Si fisso	Si fisso
Protezione inversione alim.	Si	Si	Si	No
Connettore alim.	2.1 mm plug, 5,08 mm pin or screw connector,	2.1 mm plug/pin	2.1 mm plug/pin	pin

*con scheda ausiliaria/senza scheda ausiliaria

** scheda sparkfun CC3000

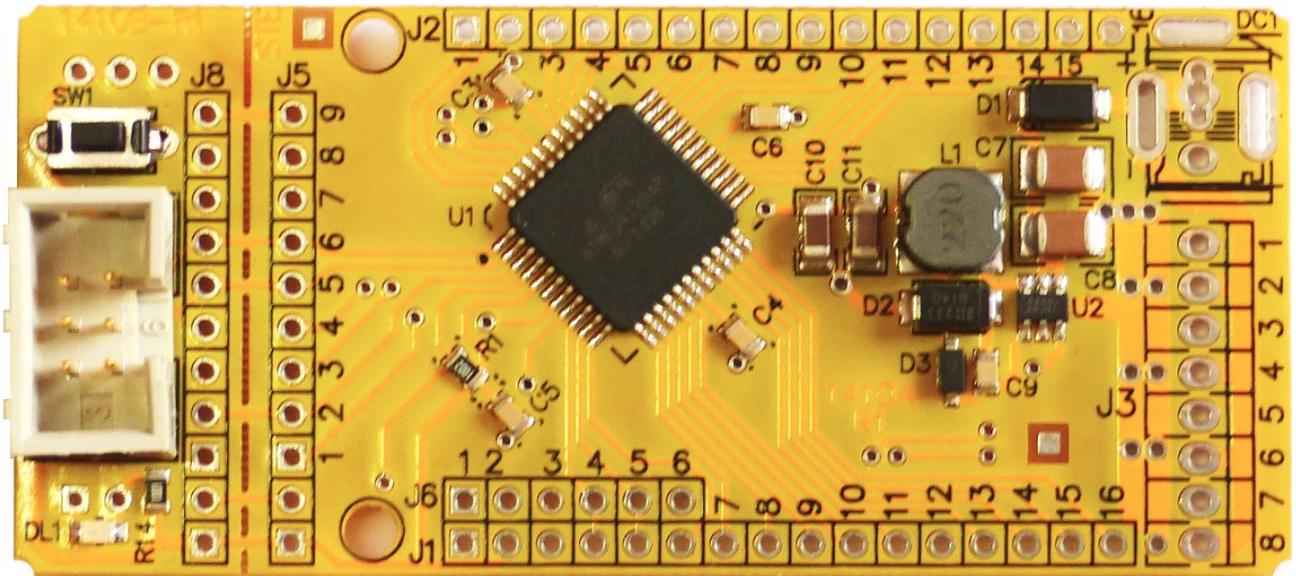
*** su scheda ausiliaria

verde: prestazione migliore

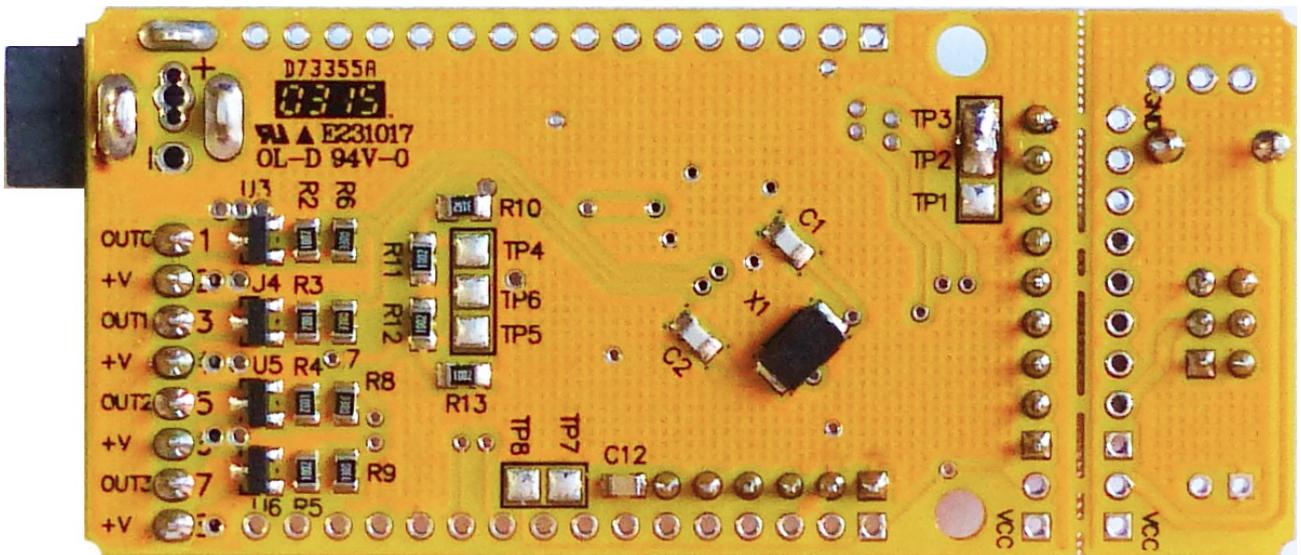
rosso: prestazione peggiore

Immagini

Fronte

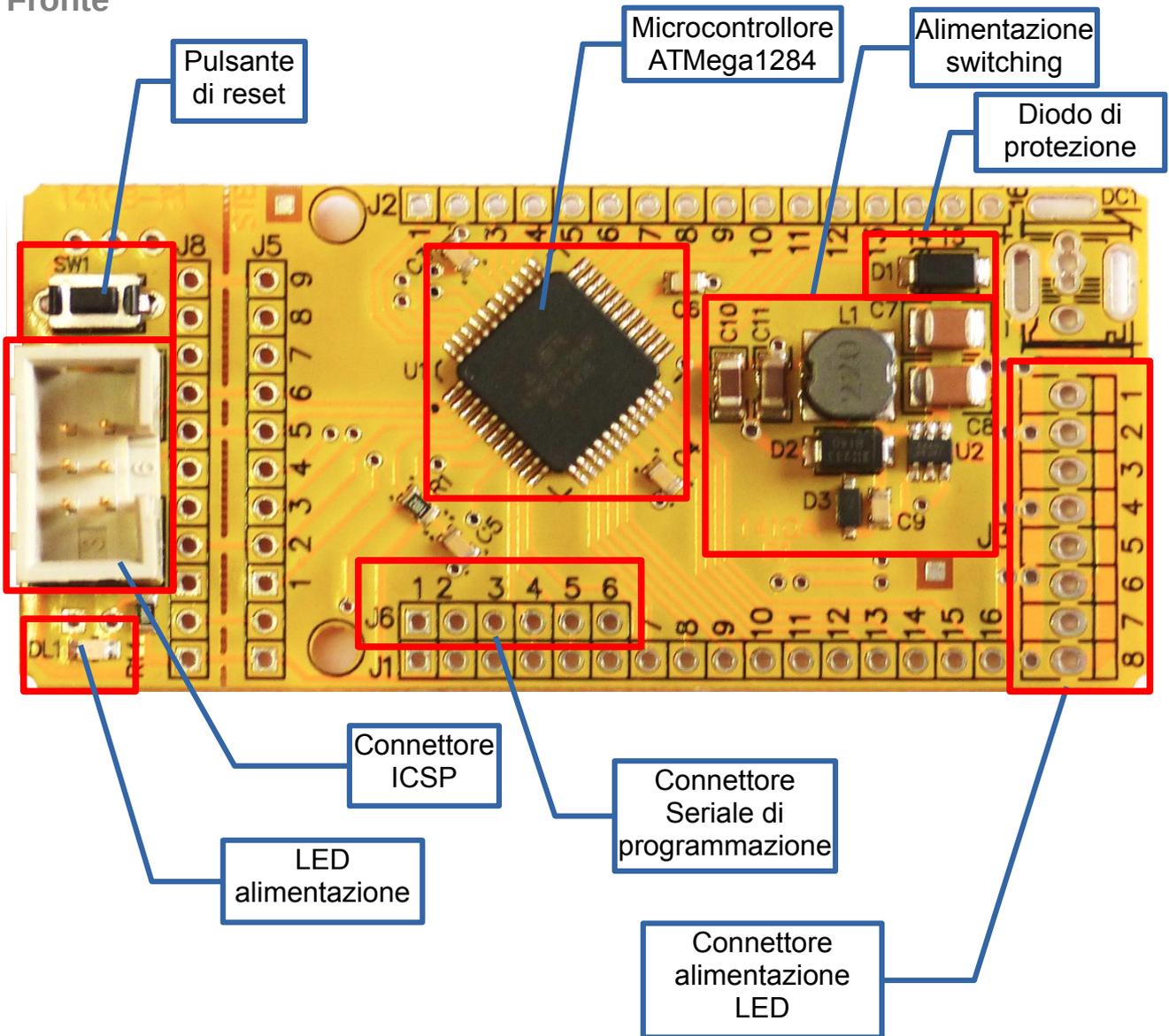


Retro

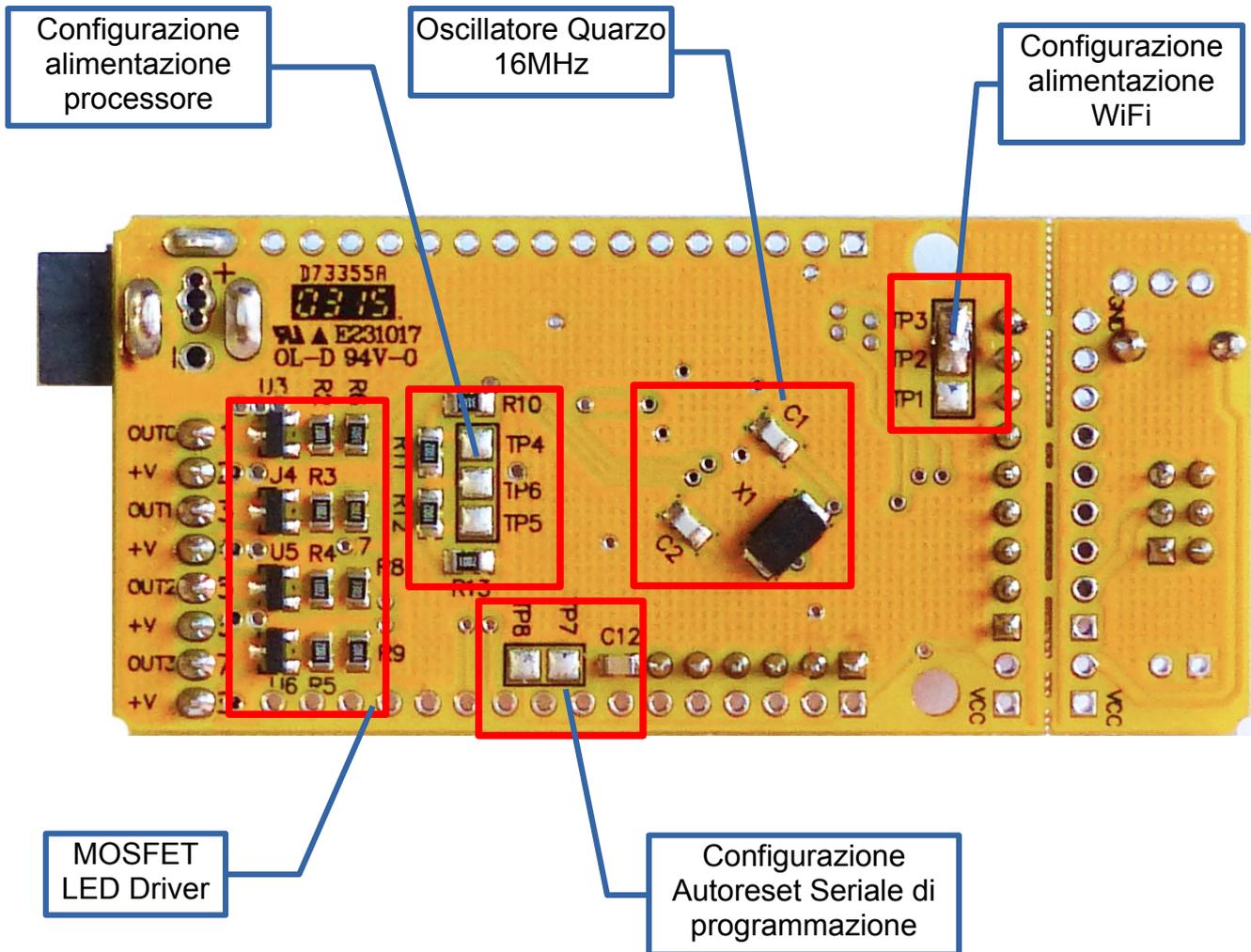


Mappa Funzionale

Fronte

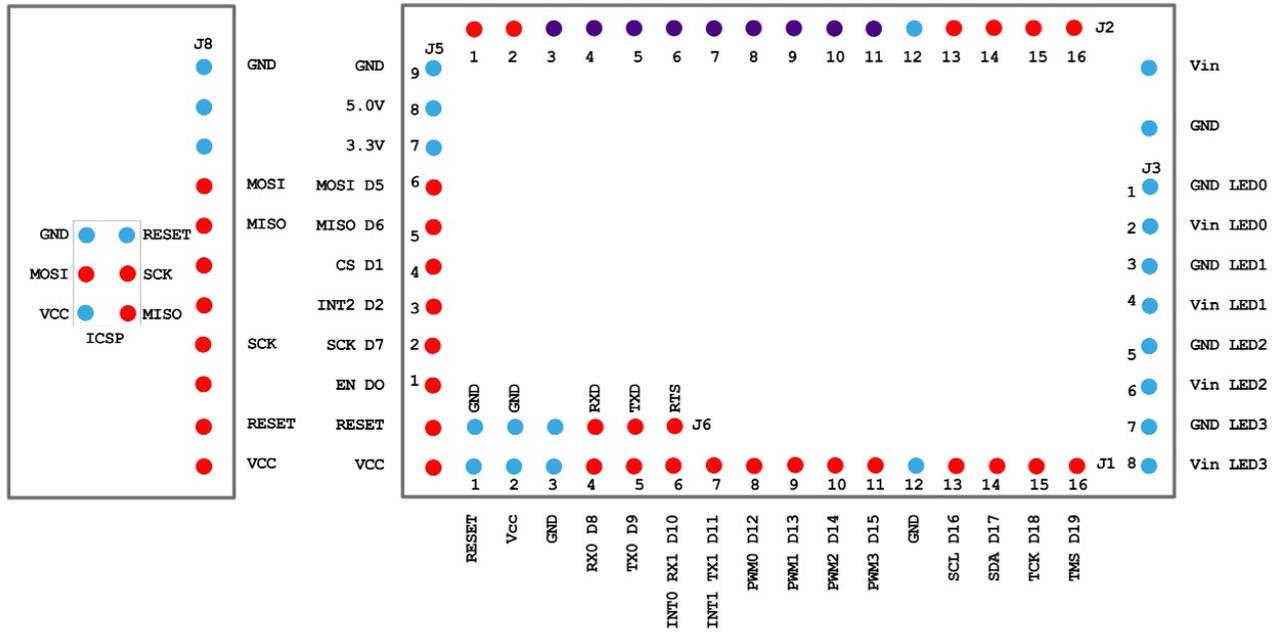


Retro



Pinout Scheda

AGEduino mini SMD Scheda Supporto



blue: alimentazione

viola: ingressi analogici / in-out digitali

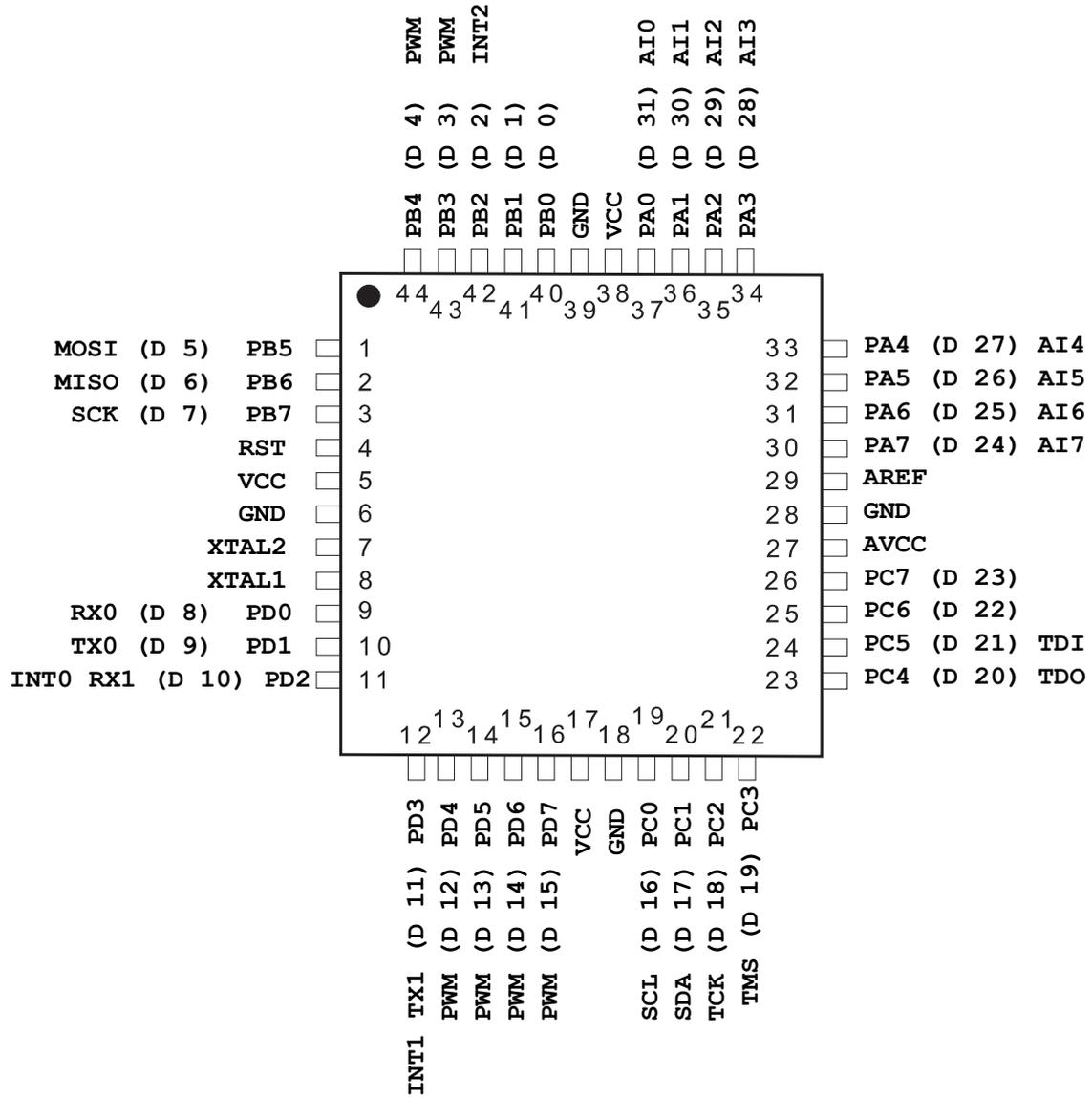
rosso: in-out digitali

Note:

tutti i contatti sono sul reticolo da 0.1" (2.54 mm)

Pin Mapping processore

Arduino			AGEduino			
Porta	Nome	Atmega328P	Porta	ATmega1284P	Pin	Conn.
D0	RX0	PD0 (PCINT16/RXD)	D8	PD0 (PCINT24/RXD0/T3)	9	J1-4
D1	TX0	PD1 (PCINT17/TXD)	D9	PD1 (PCINT25/TXD0)	10	J1-5
D2	RX1/INT0	PD2 (PCINT18/INT0)	D10	PD2 (PCINT26/RXD1/INT0)	11	J1-6
D3 pwm	TX1/INT1	PD3 (PCINT19/OC2B/INT1)	D11	PD3 (PCINT27/TXD1/INT1)	12	J1-7
D4		PD4 (PCINT20/XCK/T0)	D0	PB0 (PCINT8/XCK0/T0)	40	J5-9
D5 pwm		PD5 (PCINT21/OC0B/T1)	D1	PB1 (PCINT9/CLKO/T1)	41	J5-6
D6 pwm	INT2	PD6 (PCINT22/OC0A/AIN0)	D2	PB2 (PCINT10/INT2/AIN0)	42	J5-7
D7		PD7 (PCINT23/AIN1)	D3 pwm4	PB3 (PCINT11/OC0A/AIN1)	43	J2-2
D8		PB0 (PCINT0/CLKO/ICP1)	D14 pwm2	PD6 (PCINT30/OC2B/ICP)	15	J1-10
D9 pwm		PB1 (OC1A/PCINT1)	D13 pwm1	PD5 (PCINT29/OC1A)	14	J1-9
D10pwm		PB2 (SS/OC1B/PCINT2)	D4 pwm5	PB4 (PCINT12/OC0B/SS)	44	J2-1
D11pwm	MOSI	PB3 (MOSI/OC2A/PCINT3)	D5	PB5 (PCINT13/ICP3/MOSI)	1	J5-4
D12	MISO	PB4 (MISO/PCINT4)	D6	PB6 (PCINT14/OC3A/MISO)	2	J5-5
D13	SCK	PB5 (SCK/PCINT5)	D7	PB7 (PCINT15/OC3B/SCK)	3	J5-8
D14		-	D23	PC7 (TOSC2/PCINT23)	26	J2-13
D15		-	D22	PC6 (TOSC1/PCINT22)	25	J2-14
D16	TDI	-	D21	PC5 (TDI/PCINT21)	24	J2-15
D17	TDO	-	D20	PC4 (TDO/PCINT20)	23	J2-16
D18	TMS	-	D19	PC3 (TMS/PCINT19)	22	J1-16
D19	TCK	-	D18	PC2 (TCK/PCINT18)	21	J1-15
D20	SDA	-	D17	PC1 (SDA/PCINT17)	20	J1-14
D21	SCL	-	D16	PC0 (SCL/PCINT16)	19	J1-13
D22		-	D12 pwm0	PD4 (PCINT28/XCK1/OC1B)	13	J1-8
D23		-	D15 pwm3	PD7 (OC2A/PCINT31)	16	J1-11
A0	A0	PC0 (ADC0/PCINT8)	D31	PA0 (ADC0/PCINT0)	37	J2-3
A1	A1	PC1 (ADC1/PCINT9)	D30	PA1 (ADC1/PCINT1)	36	J2-4
A2	A2	PC2 (ADC2/PCINT10)	D29	PA2 (ADC2/PCINT2)	35	J2-5
A3	A2	PC3 (ADC3/PCINT11)	D28	PA3 (ADC3/PCINT3)	34	J2-6
A4	A4	PC4 (ADC4/PCINT12)	D27	PA4 (ADC4/PCINT4)	33	J2-7
A5	A5	PC5 (ADC5/PCINT13)	D26	PA5 (ADC5/PCINT5)	32	J2-8
A6	A6	PC6 (ADC6/PCINT14)	D25	PA6 (ADC6/PCINT6)	31	J2-9
A7	A7	PC7 (ADC7/PCINT15)	D24	PA7 (ADC7/PCINT7)	30	J2-10
Aref	Aref		Aref		29	J2-11

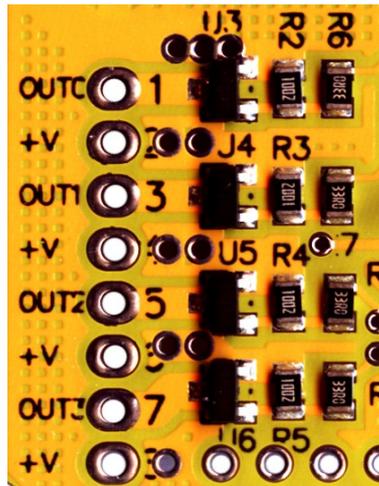


Uscite PWM transistorizzate – LED Driver

AGEduino mette a disposizione 4 transistor MOSFET IRLML2502 collegati direttamente con l'alimentazione di ingresso alla scheda (Vin).

I transistor vengono comandati dalle porte PWM0 PWM4 del processore. L'alimentazione viene portata al Drain del transistor, il Source è collegato a Ground e il pin del processore comanda il Gate tramite un resistenza di protezione, e viene collegato a Ground tramite una resistenza di pull-down.

Per collegare le strisce di LED bisogna connettere il polo positivo della striscia alla connessione Vin/+V, ed il polo negativo alla connessione GND/OUTx



La massima tensione di alimentazione è 20V, la $R_{DS(on)} = 0,04 \text{ ohm}$

La corrente massima Drain to Source è circa 3A, quindi ogni transistor può sopportare una potenza massima teorica di oltre 30 W.

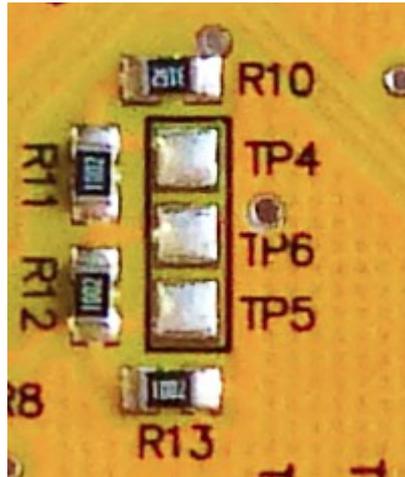
I contatti sono sul reticolo da 0.1" (2.54 mm)

Configurazione

Alimentazione processore

La scheda madre di AGEduino deve essere alimentata in corrente continua con una tensione tra i 4.5 ed i 20V.

L'alimentazione al processore può essere configurata dall'utente mettendo in corto tramite saldatura le piazzole TP4, TP5 e TP6, che si trovano sul retro della scheda madre.



Nella configurazione di default le tre piazzole non sono in corto, ed il microprocessore viene alimentato a 5Vcc .

Se vengono messe in corto le piazzole TP4 con TP6 oppure TP5 con TP6, il processore viene alimentato ad una tensione di 4 Vcc.

Se vengono messe in corto tutte e tre le piazzole (TP4-TP6-TP5) il processore viene alimentato ad una tensione di 3.3 Vcc

Anche la tensione minima di alimentazione della scheda deve essere scelta a seconda della tensione scelta per il microprocessore. Si consiglia di alimentare la scheda ad una tensione di almeno 1 volt superiore alla tensione di alimentazione del processore.

Riassumendo:

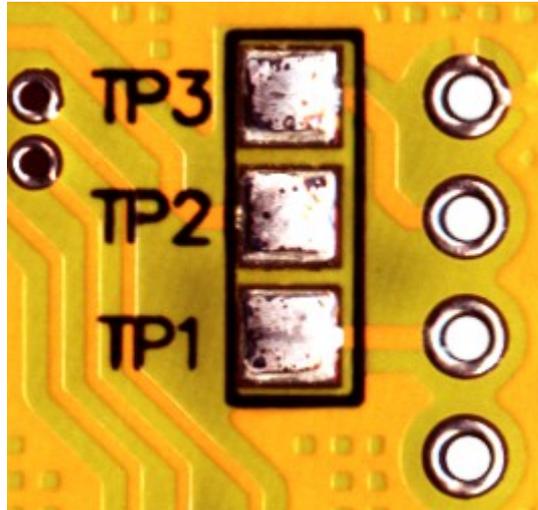
Tensione Proc.	Piazzole in corto	Tensione Alimentaz.
5V	nessuna	Da 6.5V a 20 V
4V	TP4-TP6 o TP5-TP6	Da 5.5V a 20 V
3.3V	TP4-TP6-TP5	da 4.5V a 20V

ATTENZIONE: il microprocessore a 16 Mhz e 3.3 V funziona al di fuori delle specifiche del produttore.

Alimentazione scheda WiFi

La scheda WiFi può essere alimentata direttamente a 3.3V e 4V, mentre se alimentata a 5V l'alimentazione deve avvenire attraverso il circuito di alimentazione presente sulla scheda WiFi.

Questa selezione avviene mettendo in corto i pad TP1, TP2 e TP3 posti sul retro della scheda madre vicino al connettore J5.



Se la tensione di alimentazione del processore è di 5V, allora dovranno essere messe in corto tramite saldatura le piazzole TP2 con TP3. In questo modo la tensione viene inviata al pin denominato 5V sulla scheda WiFi, quindi alimenta l'alimentatore a bordo della scheda WiFi che produce la tensione 3.3V per alimentare il chip CC3000. La tensione a 3.3V viene portata al connettore J5 al pin denominato 3.3V.

Se la tensione di alimentazione del processore è di 4V o 3.3V, allora dovranno essere messe in corto tramite saldatura le piazzole TP2 con TP1. In questo modo la tensione viene inviata al pin denominato 3.3V sulla scheda WiFi, l'alimentatore sulla scheda WiFi rimane escluso e il chip CC3000 viene alimentato direttamente alla stessa tensione del processore della scheda AGEduino.

Riassumendo:

Tensione	Piazzole in corto per processore	Piazzole in corto per WiFi
5V	nessuna	TP2-TP3
4V	TP4-TP6 o TP5-TP6	TP2-TP1
3.3V	TP4-TP6-TP5	TP2-TP1

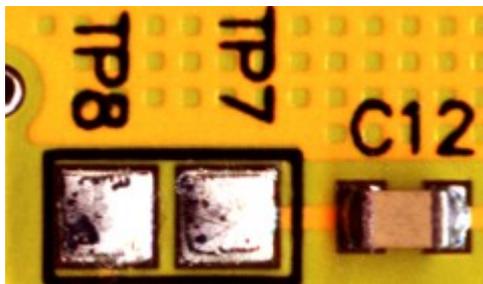
Autoreset connessione seriale

La IDE di Arduino utilizza il segnale RTS della seriale di programmazione per invocare il reset del processore prima di inviare un nuovo sketch (programma binario) alla scheda per essere registrato sulla memoria Flash. Il reset invoca il bootloader che attende i dati in arrivo dall'IDE .

Ma questo comporta che ogni volta che si inizia una connessione seriale con il processore, lo stesso viene resettato, e questo non è il funzionamento corretto se si vuole instaurare una normale connessione seriale.

AGEduino ha il circuito di autoreset, ma deve essere disconnesso di default.

Per caricare un nuovo Sketch può essere dato il reset premendo l'apposito pulsante SW1 sulla scheda di supporto appena prima dell'inizio della trasmissione, oppure possono essere messe in corto le piazzole TP7-TP8 per collegare la circuiteria di autoreset.



Queste piazzole sono poste sul retro della scheda vicino al connettore J6 .

Riassumendo:

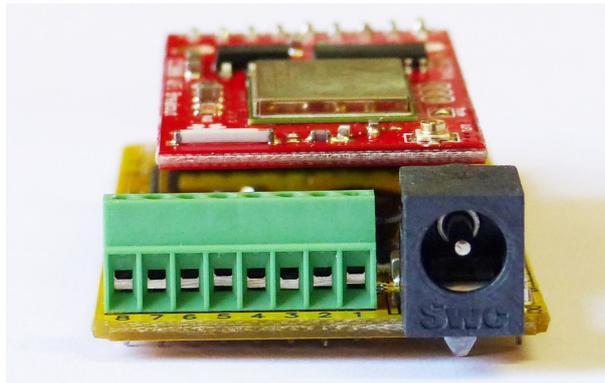
Autoreset	Piazzole
On	In corto
Off	Non corto

Connessioni

Alimentazione

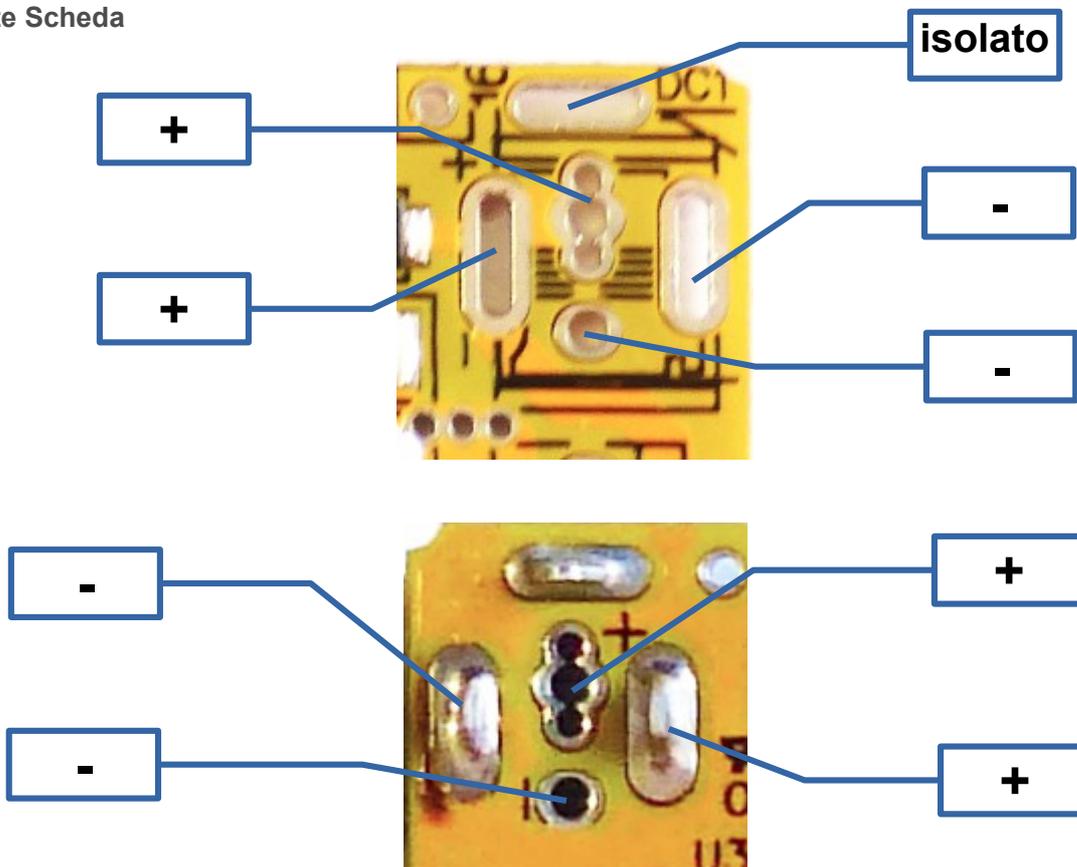
La scheda AGEduino deve essere alimentata in corrente continua con una tensione compresa tra i 4 e i 20 V. Si consiglia di utilizzare un alimentatore capace di generare una corrente di almeno 1 A, in particolare se viene connessa anche la scheda WiFi.

AGEduino Model B monta un connettore di alimentazione coassiale da 2,1 mm, identico a quello utilizzato da Arduino UNO. Il polo positivo è sul connettore centrale.

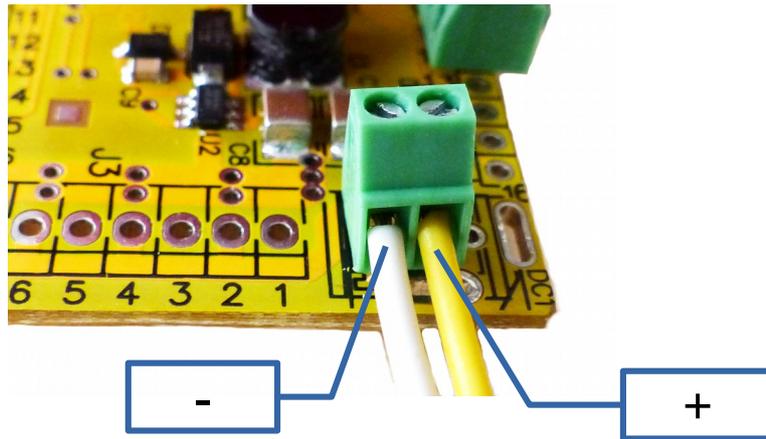


AGEduino Model A permette sia la connessione diretta che il montaggio del connettore coassiale come del connettore a vite.

Fronte Scheda

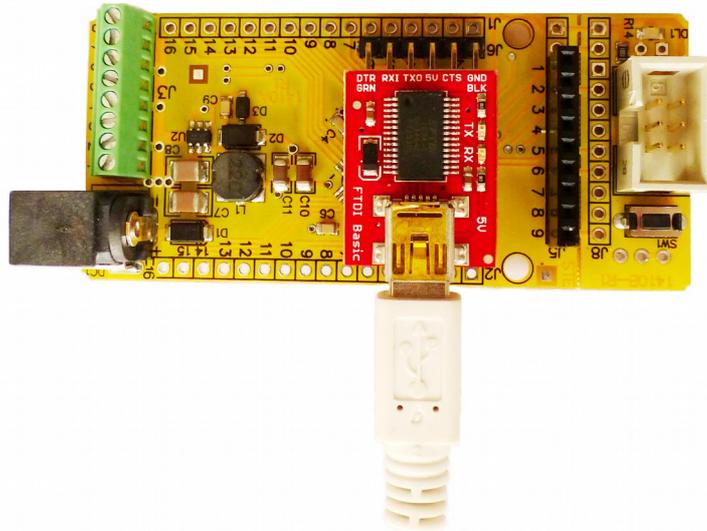


Connettore a vite passo 0.2" (5.08 mm)



AGEduino può essere alimentato tramite **batteria** con tensione maggiore di 4.5 Vcc.

Seriale di Programmazione



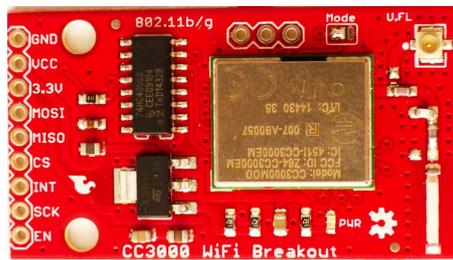
Il contatto denominato 5V è isolato in AGEduino quindi:

- il processore non viene alimentato dalla seriale
- la scheda deve essere alimentata esternamente per essere programmata
- si può utilizzare il convertitore a 5V qualsiasi sia la tensione di alimentazione selezionata per il processore

Il contatto denominato DTR viene utilizzato dalla IDE di Arduino per resettare in automatico il processore e richiamare la procedura di bootloader per il caricamento dello Sketch (programma) nella memoria Flash del processore. Di default questa connessione non è abilitata. Per abilitarla vedere il capitolo Autoreset connessione seriale.

Scheda WiFi

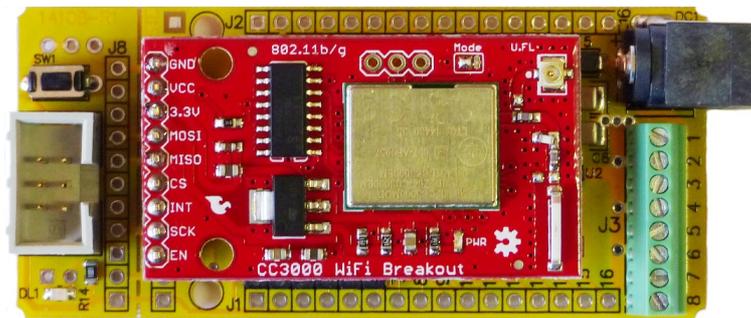
AGEduino è predisposto per la connessione alla scheda WiFi “CC3000 breakout” di Sparkfun.



La scheda WiFi deve essere collegata tramite il connettore J5 presente sulla scheda AGEduino.

Su AGEduino model B è montato un connettore femmina 9 poli. Per eseguire la connessione bisogna saldare sulla scheda WiFi un connettore maschio 9 poli.

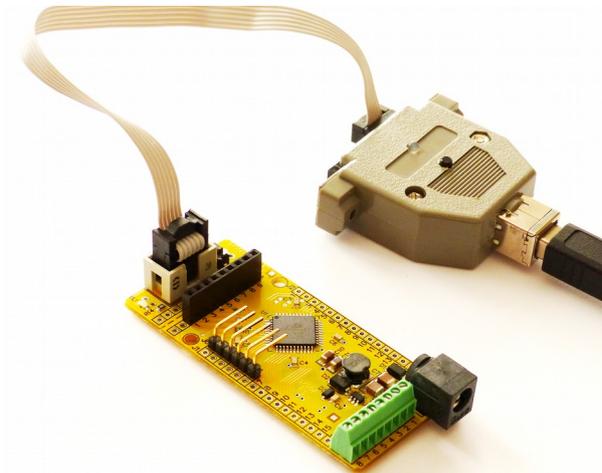
Per essere alimentata, deve essere configurata la corretta alimentazione tramite corto sulle piazzole come descritto nel capitolo Alimentazione scheda WiFi .



Connessione ICSP

Per programmare il microprocessore tramite interfaccia ICSP bisogna utilizzare un apposito programmatore. Nell'immagine possiamo vedere il programmatore USB per ATMEL AVR-ISP500-TINY di OLIMEX.

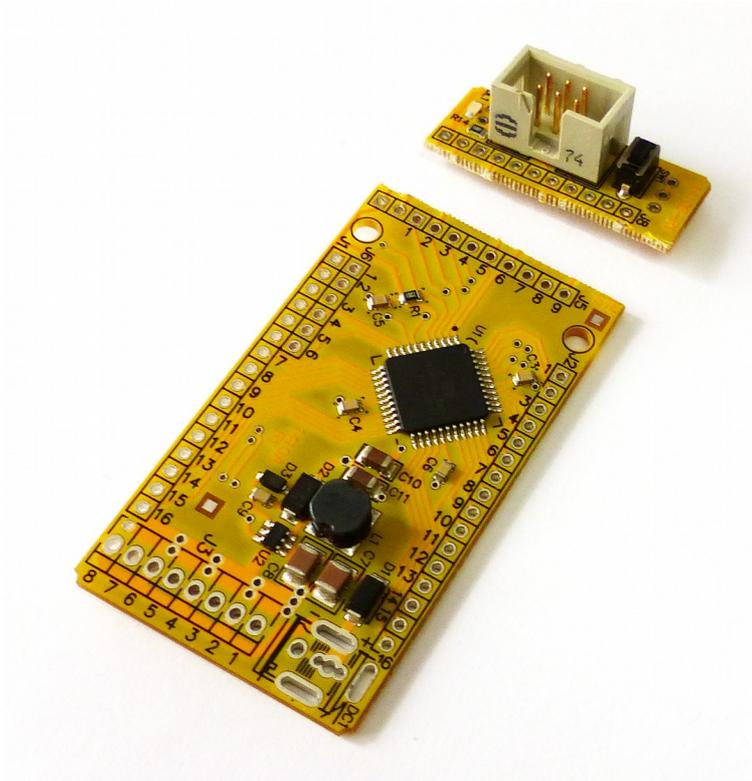
Perchè la programmazione abbia successo deve essere collegata l'alimentazione della scheda AGEduino.



Rimozione Scheda Supporto

La scheda di supporto può essere facilmente rimossa con l'aiuto di un tronchesino per elettronica.

Non flettere eccessivamente le schede per non danneggiare le stesse.



É sempre possibile ricollegare la scheda di supporto montando opportuni connettori su J5 e J8 . Devono essere collegati anche i due pin posti prima del connettore J5 e J8 che provvedono l'alimentazione e il reset, si consiglia quindi di installare strip a 11 poli.

Configurazione IDE Arduino

AGEduino è compatibile con la IDE (Integrated Development Environment) di Arduino tramite le librerie di supporto del progetto Sanguino. Attualmente viene supportata da AGEvoluzione solo la **IDE 1.0.5**

Per installare la IDE bisogna scaricarla dal sito ufficiale di Arduino: <http://arduino.cc/en/Main/Software>

Una volta terminata l'installazione e creata la cartella sketchbook bisogna scaricare la libreria Sanguino mantenuta su GitHub da Kristian Lauszus (<https://github.com/Lauszus/Sanguino>). L'ultima release può essere scaricata direttamente da questo link: <https://github.com/Lauszus/Sanguino/archive/master.zip>

Decomprimere il file zip nella directory sketchbook/hardware e rinominare la cartella "Sanguino-master" in "Sanguino" (sketchbook/hardware/Sanguino).

Nella IDE di Arduino selezionare il processore "Sanguino W/ ATmega1284 or ATmega1284P (16MHz)" dal menù "Strumenti" → "Tipo di Arduino".

AGEduino viene fornito con il Bootloader Optiboot già installato nella memoria flash, quindi basta connettere AGEduino all'alimentazione e alla USB del computer tramite l'opportuno cavo e adattatore USB-Seriale per programmare il dispositivo.

Per ulteriori informazioni consultare i capitoli:

- Alimentazione processore
- Autoreset connessione seriale
- Seriale di Programmazione

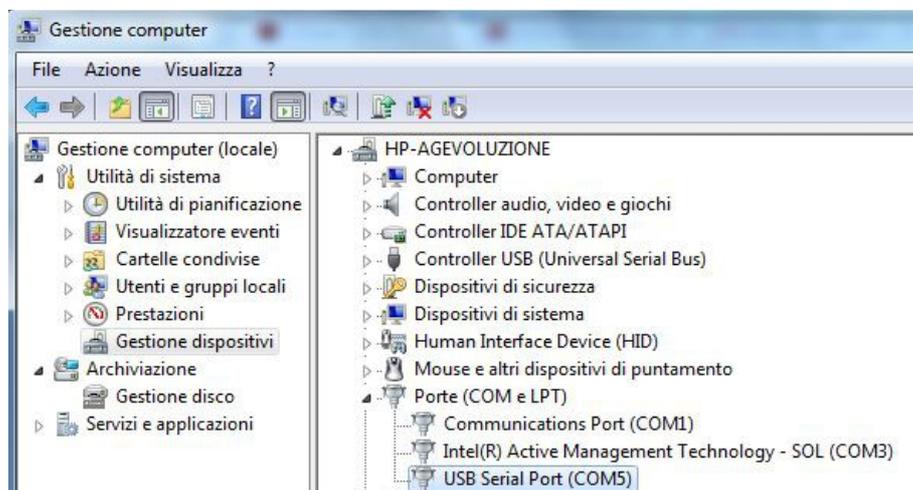
Per i sistemi operativi linux x può essere necessario aggiungere l'utente che utilizza l'IDE al gruppo dialout o al gruppo uucp.

I sistemi operativi Linux non necessitano di ulteriore software, invece **su sistemi operativi Windows** bisogna installare il driver USB per il chip FT232R di FTDI dal loro sito:

<http://www.ftdichip.com/FTDrivers.htm>

Selezionare il dispositivo corretto ed installare il driver prima di attivare l'IDE .

Se l'installazione è avvenuta in modo corretto, una nuova porta di comunicazione apparirà nella Gestione Computer di Windows



Prima inizializzazione del processore

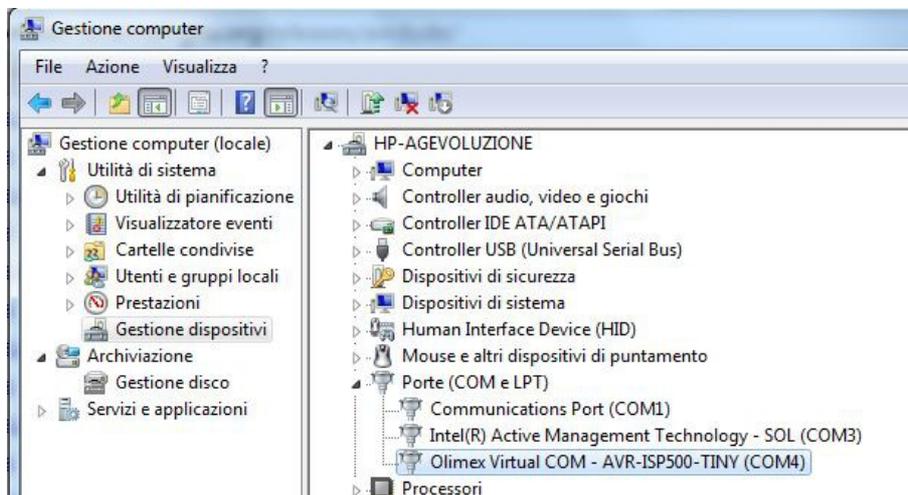
AGEduino viene fornito già inizializzato per il funzionamento a 16 Mhz e con il bootloader Optiboot installato. Nel caso si presenti il bisogno di reinizializzare o riconfigurare il dispositivo seguire le istruzioni di questo capitolo

L'inizializzazione del processore viene condotta utilizzando il programmatore AVR-ISP500-TINY di OLIMEX tramite connettore ICSP ed utilizzando il programma AVRdude per leggere/scrivere lo stato del processore.

Su **sistemi operativi Windows** deve essere installato il driver di comunicazione che può essere scaricato dal sito del produttore:

<https://www.olimex.com/Products/AVR/Programmers/AVR-ISP500-TINY/>

Se l'installazione è avvenuta in modo corretto, una nuova porta di comunicazione apparirà nella Gestione Computer di Windows



Una volta connesso il programmatore a connettore ICSP e connessa alimentazione di AGEDuino, siamo pronti per la prima comunicazione con il processore.

La prima configurazione avviene attraverso la configurazione dei cosiddetti “fuse” del microprocessore.

Per calcolare il valore dei fuse è utile utilizzare il “fuse calculator” disponibile su <http://eleccelerator.com/fusecalc/>.

Per la prima connessione o in caso sia stata scritta una configurazione errata è importante utilizzare il parametro -B 4 che forza la comunicazione a bassa velocità.

I comandi nelle pagine seguenti sono nel formato per Linux. Su sistemi operativi Windows il comando avrdude deve essere completato con il percorso:

```
ARDUINO_INST_DIR\hardware\tools\avr\bin\avrdude.exe
```

e la porta seriale deve essere indicata come -P //./COM4

Con questo primo comando verifichiamo che la connessione sia presente e configurata correttamente:

```
avrdude -p ATmega1284P -c stk500v2 -P /dev/ttyACM0 -v -B 4
```

```
avrdude: Version 6.1, compiled on Jun 14 2014 at 07:40:00
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2014 Joerg Wunsch

System wide configuration file is "/etc/avrdude/avrdude.conf"
User configuration file is "/home/umbi/.avrduderc"
User configuration file does not exist or is not a regular file, skipping

Using Port                : /dev/ttyACM0
Using Programmer          : stk500v2
Setting bit clk period   : 4.0
AVR Part                  : ATmega1284P
Chip Erase delay         : 55000 us
PAGEL                     : PD7
BS2                       : PA0
RESET disposition        : dedicated
RETRY pulse              : SCK
serial program mode      : yes
parallel program mode    : yes
Timeout                  : 200
StabDelay                : 100
CmdexeDelay              : 25
SyncLoops                : 32
ByteDelay                : 0
PollIndex                : 3
PollValue                : 0x53
Memory Detail            :

                Block Poll
Memory Type Mode Delay Size  Indx Paged  Size   Size #Pages MinW  MaxW   Polled
-----
eeprom        65    10   128    0 no    4096    8     0   9000   9000 0xff 0xff
flash         65    10   256    0 yes   131072 256   512  4500  4500 0xff 0xff
lock           0     0     0     0 no     1     0     0   9000   9000 0x00 0x00
lfuse          0     0     0     0 no     1     0     0   9000   9000 0x00 0x00
hfuse          0     0     0     0 no     1     0     0   9000   9000 0x00 0x00
efuse          0     0     0     0 no     1     0     0   9000   9000 0x00 0x00
signature      0     0     0     0 no     3     0     0     0     0 0x00 0x00
calibration    0     0     0     0 no     1     0     0     0     0 0x00 0x00

Programmer Type : STK500V2
Description    : Atmel STK500 Version 2.x firmware
Programmer Model: STK500
Hardware Version: 2
Firmware Version Master : 2.10
Topcard       : Unknown
Vtarget       : 3.3 V
SCK period    : 8.7 us
Varef         : 3.3 V
Oscillator    : 60.433 kHz
```

```
avrdude: AVR device initialized and ready to accept instructions
```

```
Reading | ##### | 100% 0.01s
```

```
avrdude: Device signature = 0x1e9705
avrdude: safemode: lfuse reads as 62
avrdude: safemode: hfuse reads as 99
avrdude: safemode: efuse reads as FF

avrdude: safemode: lfuse reads as 62
avrdude: safemode: hfuse reads as 99
avrdude: safemode: efuse reads as FF
avrdude: safemode: Fuses OK (E:FF, H:99, L:62)

avrdude done. Thank you.
```

Se l'output del programma corrisponde a questo allora la connessione è corretta ed AGEduino sta funzionando correttamente.

Bisogna quindi configurare i fuse per utilizzare il clock esterno a 16 Mhz:

```
avrdude -p ATmega1284P -c stk500v2 -U hfuse:w:0x9a:m -U lfuse:w:0xff:m -P /dev/ttyACM0 -B 4
```

```
avrdude: AVR device initialized and ready to accept instructions
```

```
Reading | ##### | 100% 0.01s
```

```
avrdude: Device signature = 0x1e9705
```

```
avrdude: reading input file "0x9a"
```

```
avrdude: writing hfuse (1 bytes):
```

```
Writing | ##### | 100% 0.02s
```

```
avrdude: 1 bytes of hfuse written
```

```
avrdude: verifying hfuse memory against 0x9a:
```

```
avrdude: load data hfuse data from input file 0x9a:
```

```
avrdude: input file 0x9a contains 1 bytes
```

```
avrdude: reading on-chip hfuse data:
```

```
Reading | ##### | 100% 0.00s
```

```
avrdude: verifying ...
```

```
avrdude: 1 bytes of hfuse verified
```

```
avrdude: reading input file "0xff"
```

```
avrdude: writing lfuse (1 bytes):
```

```
Writing | ##### | 100% 0.01s
```

```
avrdude: 1 bytes of lfuse written
```

```
avrdude: verifying lfuse memory against 0xff:
```

```
avrdude: load data lfuse data from input file 0xff:
```

```
avrdude: input file 0xff contains 1 bytes
```

```
avrdude: reading on-chip lfuse data:
```

```
Reading | ##### | 100% 0.00s
```

```
avrdude: verifying ...
```

```
avrdude: 1 bytes of lfuse verified
```

```
avrdude: safemode: Fuses OK (E:FF, H:9A, L:FF)
```

```
avrdude done. Thank you.
```

Per avere ulteriore conferma che la programmazione dei fuse ha avuto successo si può richiedere lo stato del processore senza il flag -B 4

```
avrdude -p ATmega1284P -c stk500v2 -P /dev/ttyACM0 -v
```

```
avrdude: Version 6.1, compiled on Jun 14 2014 at 07:40:00
```

```
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
```

```
Copyright (c) 2007-2014 Joerg Wunsch
```

```
System wide configuration file is "/etc/avrdude/avrdude.conf"
```

```
User configuration file is "/home/umbi/.avrduderc"
```

```
User configuration file does not exist or is not a regular file, skipping
```

```
Using Port          : /dev/ttyACM0
Using Programmer    : stk500v2
AVR Part           : ATmega1284P
Chip Erase delay   : 55000 us
PAGEL              : PD7
```

```

BS2 : PA0
RESET disposition : dedicated
RETRY pulse : SCK
serial program mode : yes
parallel program mode : yes
Timeout : 200
StabDelay : 100
CmdexeDelay : 25
SyncLoops : 32
ByteDelay : 0
PollIndex : 3
PollValue : 0x53
Memory Detail :

```

Memory	Type	Mode	Delay	Block Size	Poll Indx	Paged	Size	Page Size	#Pages	MinW	MaxW	Polled	
												ReadBack	
eeeprom		65	10	128	0	no	4096	8	0	9000	9000	0xff	0xff
flash		65	10	256	0	yes	131072	256	512	4500	4500	0xff	0xff
lock		0	0	0	0	no	1	0	0	9000	9000	0x00	0x00
lfuse		0	0	0	0	no	1	0	0	9000	9000	0x00	0x00
hfuse		0	0	0	0	no	1	0	0	9000	9000	0x00	0x00
efuse		0	0	0	0	no	1	0	0	9000	9000	0x00	0x00
signature		0	0	0	0	no	3	0	0	0	0	0x00	0x00
calibration		0	0	0	0	no	1	0	0	0	0	0x00	0x00

```

Programmer Type : STK500V2
Description : Atmel STK500 Version 2.x firmware
Programmer Model: STK500
Hardware Version: 2
Firmware Version Master : 2.10
Topcard : Unknown
Vtarget : 3.3 V
SCK period : 8.7 us
Varef : 3.3 V
Oscillator : 60.433 kHz

```

avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.01s

```

avrdude: Device signature = 0x1e9705
avrdude: safemode: lfuse reads as FF
avrdude: safemode: hfuse reads as 9A
avrdude: safemode: efuse reads as FF

avrdude: safemode: lfuse reads as FF
avrdude: safemode: hfuse reads as 9A
avrdude: safemode: efuse reads as FF
avrdude: safemode: Fuses OK (E:FF, H:9A, L:FF)

avrdude done. Thank you.

```

Installazione supporto ICSP AVR-ISP500-TINY

Questo programmatore non è supportato nativamente dall'IDE di Arduino, ma con due semplici passaggi è possibile abilitare il supporto.

1. Creare la directory Olimex in sketchbook/hardware. Quindi creare il file programmers.txt nella directory creata (sketchbook/hardware/Olimex) e inserire le seguenti righe:

```
olimexisp.name=AVR ISP 500 Olimex  
olimexisp.communication=serial  
olimexisp.protocol=stk500v2
```

2. Aprire il file sketchbook/hardware/Sanguino/boards.txt ed aggiungere in fondo al testo presente le seguenti righe:

```
#####
```

```
atmega1284p_olimex.name=Sanguino W/ ATmega1284/P Olimex (16MHz)
```

```
atmega1284p_olimex.upload.using=olimexisp  
atmega1284p_olimex.upload.maximum_size=131072  
atmega1284p_olimex.upload.maximum_data_size=16384
```

```
atmega1284p_olimex.upload.speed=57600  
atmega1284p_olimex.bootloader.path=optiboot  
atmega1284p_olimex.bootloader.file=optiboot_atmega1284p.hex
```

```
atmega1284p_olimex.bootloader.low_fuses=0xFF  
atmega1284p_olimex.bootloader.high_fuses=0xDE  
atmega1284p_olimex.bootloader.extended_fuses=0xFD  
atmega1284p_olimex.bootloader.unlock_bits=0x3F  
atmega1284p_olimex.bootloader.lock_bits=0x0F
```

```
atmega1284p_olimex.build.mcu=atmega1284p  
atmega1284p_olimex.build.f_cpu=1600000L  
atmega1284p_olimex.build.board=AVR_SANGUINO  
atmega1284p_olimex.build.core=arduino:arduino  
atmega1284p_olimex.build.variant=sanguino
```

```
#####
```

ATTENZIONE: questo dispositivo deve essere selezionato nell'IDE di Arduino tramite "Strumenti" → "Scheda" solo se si vuole scrivere direttamente lo sketch (programma) tramite programmatore Olimex senza utilizzare il Bootloader Optiboot.

Installazione Bootloader

Per installare il bootloader Optiboot fornito con le librerie Sanguino bisogna:

1. collegare il programmatore al connettore ICSP
2. connettere alimentazione di AGEduino
3. attivare la IDE di Arduino.
4. Quindi bisogna selezionare la corretta porta seriale tramite il menu “Strumenti” → “Porta Seriale” e quindi il programmatore “AVR ISP 500 Olimex” tramite il menu “Strumenti” → “Programmatore”.

Per scrivere il bootloader selezionare il menu “Strumenti” → “Scrivi il bootloader”.

Programmazione tramite ICSP

Lo sketch (programma) creato può essere scritto sulla memoria Flash del processore tramite connessione ICSP. In questo caso non viene utilizzato il bootloader di AGEduino.

Oltre a collegare il programmatore ICSP ed installarlo nell'IDE come descritto in:

- Connessione ICSP
- Installazione supporto ICSP AVR-ISP500-TINY

bisogna configurare correttamente l'IDE:

1. tramite il menù “Strumenti” → “Tipo di Arduino” selezionare “Sanguino W/ ATmega1284/P Olimex (16MHz)”
2. tramite il menù “Strumenti” → “Porta Seriale” selezionare il corretto dispositivo seriale
3. tramite il menu “Strumenti” → “Programmatore” selezionare “AVR ISP 500 Olimex”

Siamo pronti per scaricare lo sketch tramite connessione ICSP.

Programmazione Tramite USB/Seriale

Lo sketch (programma) creato può essere scritto sulla memoria Flash del processore tramite connessione seriale collegata alla seriale di programmazione. AGEduino dispone di 2 seriali asincrone, ma solo una, la seriale 0, può essere utilizzata per scrivere il programma. In questo caso viene utilizzato il bootloader di AGEduino.

Oltre collegare il computer alla seriale di AGEduino come descritto in:

- Autoreset connessione seriale
- Seriale di Programmazione

bisogna configurare correttamente l'IDE:

1. tramite il menù “Strumenti” → “Tipo di Arduino” selezionare “Sanguino W/ ATmega1284 or ATmega1284P (16MHz)”
2. tramite il menù “Strumenti” → “Porta Seriale” selezionare il corretto dispositivo seriale

3. tramite il menu “Strumenti” → “Programmatore” selezionare “AVRISP mkII”

Siamo pronti per scaricare lo sketch tramite connessione USB/Seriale. **Ricordo che se la circuiteria di autoreset non è stata collegata, per caricare lo sketch su AGEduino bisogna premere il pulsante di reset (SW1) immediatamente prima dell'invio dello sketch.**

Supporto scheda WiFi

La scheda WiFi CC3000 non è supportata nativamente dall'IDE di Arduino. Per poterla utilizzare bisogna scaricare le apposite librerie di interfaccia.

AGEvoluzione supporta le librerie Adafruit_CC3000 di Adafruit disponibile su GitHub (https://github.com/adafruit/Adafruit_CC3000_Library) e la cui più recente release può essere scaricata da: https://github.com/adafruit/Adafruit_CC3000_Library/archive/master.zip

Decomprimere il file zip nella directory sketchbook/libraries e rinominare la cartella “Adafruit_CC3000_Library-master” in “Adafruit_CC3000” (sketchbook/libraries/Adafruit_CC3000).

A questo punto si possono testare gli Sketch di esempio forniti da Adafruit, previo la riconfigurazione della mappatura dei pin di controllo della scheda nel codice sorgente dello sketch tramite le direttive #define :

```
#define ADAFRUIT_CC3000_IRQ 2
#define ADAFRUIT_CC3000_VBAT 0
#define ADAFRUIT_CC3000_CS 1
```

Ricordo che prima di utilizzare la scheda WiFi si deve procedere alla selezione della sorgente di alimentazione come descritto nel capitolo “Alimentazione scheda WiFi”

Programmi di Esempio

Installazione IDE 1.6.3 Sperimentale

La compilazione funziona anche con IDE 1.6.3, ma bisogna fare alcune modifiche alla directory Sanguino in /Arduino/hardware scaricata da github di Lauszus.

1. Nella directory Sanguino cancellare tutti i files e le directory tranne avr
2. Selezionare la scheda Sanguino nell'IDE e quindi il processore (menù strumenti)

Per il programmatore Olimex bisogna mettere il file programmers.txt nella directory Arduino/hardware/Sanguino/avr/ con il seguente contenuto:

```
olimexisp.name=AVR ISP 500 Olimex
olimexisp.communication=serial
olimexisp.protocol=stk500v2
olimexisp.program.protocol=stk500v2
olimexisp.program.tool=avrdude
olimexisp.program.extra_params=-P{serial.port}
```

e aggiungere a boards.txt (ATTENZIONE! Diverso da aggiunta per IDE 1.0.5)

```
#####

sanguino.menu.cpu.atmega1284p_olimex=ATmega1284/P Olimex (16MHz)

sanguino.menu.cpu.atmega1284p_olimex.upload.using=olimexisp
sanguino.menu.cpu.atmega1284p_olimex.upload.maximum_size=131072
sanguino.menu.cpu.atmega1284p_olimex.upload.maximum_data_size=16384

sanguino.menu.cpu.atmega1284p_olimex.upload.speed=57600
sanguino.menu.cpu.atmega1284p_olimex.bootloader.path=optiboot
sanguino.menu.cpu.atmega1284p_olimex.bootloader.file=optiboot_atmega1284p.he
x

sanguino.menu.cpu.atmega1284p_olimex.bootloader.low_fuses=0xFF
sanguino.menu.cpu.atmega1284p_olimex.bootloader.high_fuses=0xDE
sanguino.menu.cpu.atmega1284p_olimex.bootloader.extended_fuses=0xFD
sanguino.menu.cpu.atmega1284p_olimex.bootloader.unlock_bits=0x3F
sanguino.menu.cpu.atmega1284p_olimex.bootloader.lock_bits=0x0F

sanguino.menu.cpu.atmega1284p_olimex.build.mcu=atmega1284p
sanguino.menu.cpu.atmega1284p_olimex.build.f_cpu=1600000L
sanguino.menu.cpu.atmega1284p_olimex.build.board=AVR_SANGUINO
sanguino.menu.cpu.atmega1284p_olimex.build.core=arduino:arduino
sanguino.menu.cpu.atmega1284p_olimex.build.variant=sanguino

#####
```

La nuova board appare nel menù “Strumenti” → “Processore”

Installazione IDE 1.7.4 Sperimentale

La compilazione funziona anche con IDE 1.7.4, ma bisogna fare alcune modifiche alla directory Sanguino in /Arduino/hardware scaricata da github di Lauszus.

3. Nella directory Sanguino cancellare tutti i files e le directory tranne avr
4. Selezionare la scheda Sanguino nell'IDE e quindi il processore (menù strumenti)

Per il programmatore Olimex bisogna mettere il file programmers.txt nella directory Arduino/hardware/Sanguino/avr/ con il seguente contenuto:

```
olimexisp.name=AVR ISP 500 Olimex
olimexisp.communication=serial
olimexisp.protocol=stk500v2
olimexisp.program.protocol=stk500v2
olimexisp.program.tool=avrdude
olimexisp.program.extra_params=-P{serial.port}
```

e aggiungere a boards.txt (ATTENZIONE! Diverso da aggiunta per IDE 1.0.5)

```
#####

sanguino.menu.cpu.atmega1284p_olimex=ATmega1284/P Olimex (16MHz)

sanguino.menu.cpu.atmega1284p_olimex.upload.using=olimexisp
sanguino.menu.cpu.atmega1284p_olimex.upload.maximum_size=131072
sanguino.menu.cpu.atmega1284p_olimex.upload.maximum_data_size=16384

sanguino.menu.cpu.atmega1284p_olimex.upload.speed=57600
sanguino.menu.cpu.atmega1284p_olimex.bootloader.path=optiboot
sanguino.menu.cpu.atmega1284p_olimex.bootloader.file=optiboot_atmega1284p.he
x

sanguino.menu.cpu.atmega1284p_olimex.bootloader.low_fuses=0xFF
sanguino.menu.cpu.atmega1284p_olimex.bootloader.high_fuses=0xDE
sanguino.menu.cpu.atmega1284p_olimex.bootloader.extended_fuses=0xFD
sanguino.menu.cpu.atmega1284p_olimex.bootloader.unlock_bits=0x3F
sanguino.menu.cpu.atmega1284p_olimex.bootloader.lock_bits=0x0F

sanguino.menu.cpu.atmega1284p_olimex.build.mcu=atmega1284p
sanguino.menu.cpu.atmega1284p_olimex.build.f_cpu=1600000L
sanguino.menu.cpu.atmega1284p_olimex.build.board=AVR_SANGUINO
sanguino.menu.cpu.atmega1284p_olimex.build.core=arduino:arduino
sanguino.menu.cpu.atmega1284p_olimex.build.variant=sanguino

#####
```

La nuova board appare nel menù “Strumenti” → “Processore”