

# ADC DAC GPIO Board for Raspberry

Patrick Predella 165283, Federico D'Eredità 151646

## Indice

1	Introduzione	1
2	Componentistica	2
2.1	Circuiteria Microchip utilizzata . . . . .	2
2.1.1	ADC - Analog-to-Digital Converter . . . . .	2
2.1.2	DAC - Digital-to-Analog Converter . . . . .	2
2.1.3	GP I/O - General Purpose Input/Output Device . . . . .	2
2.2	Componenti Hardware . . . . .	2
2.2.1	La scheda Piggy-Back e il DCDC Converter . . . . .	2
2.2.2	Il Raspberry . . . . .	2
2.3	Il linguaggio di programmazione I2C . . . . .	2
3	Realizzazione del Progetto	2
3.1	Protezione della componentistica . . . . .	2
3.1.1	Filtri RC . . . . .	2
3.1.2	Protezioni V e I . . . . .	3
3.2	Schema elettrico d'assemblaggio . . . . .	3

## 1 Introduzione

Finalità del progetto è il monitoraggio di misure di temperatura e pressione ambientali mediante sensoristica analogica con la possibilità di comandare attuatori analogici e/o digitali tramite programmazione di un Raspberry Pi2+. A questo scopo abbiamo deciso di realizzare una scheda composta dai seguenti elementi principali:

- Un convertitore analogico-digitale per il campionamento dei segnali analogici di temperatura e pressione affiancato da una protezione per eventuali sbalzi di tensione;
- Un convertitore digitale-analogico per il controllo di attuatori analogici anch'esso affiancato da una protezione per eventuali sbalzi di tensione;
- Un general purpose input/output per il controllo di attuatori digitali ed il controllo delle operazioni pre-programmate nel Raspberry;
- Scheda PiggyBack (vedasi capitolo 2.2.1) per l'assemblaggio della componentistica;
- Un Raspberry Pi2+ per il controllo delle linee dati, di clock e l'interazione con il GPI/O.

## 2 Componentistica

### 2.1 Ciuiteria Microchip utilizzata

#### 2.1.1 ADC - Analog-to-Digital Converter

#### 2.1.2 DAC - Digital-to-Analog Converter

#### 2.1.3 GP I/O - General Purpose Input/Output Device

### 2.2 Componenti Hardware

#### 2.2.1 La scheda Piggy-Back e il DCDC Converter

#### 2.2.2 Il Raspberry

### 2.3 Il linguaggio di programmazione I2C

## 3 Realizzazione del Progetto

Il progetto è molto semplice. Si tratta di:

- utilizzare il DCDC converter per stabilizzare l'alimentazione dei componenti.
- portare le piste a degli header in modo che siano accessibili al Raspberry
- proteggere le linee in ingresso e in uscita in modo che non si danneggino i componenti

### 3.1 Protezione della componentistica

Si tratta di rispettare le specifiche di absolute maximum ratings, proteggere le linee in ingresso e in uscita da ingressi **out of range** e **ridurre i rumori**. In particolare un filtro per i rumori nel caso dell'ADC, un filtro per avere un transitorio non nullo nel caso dei GPIO e in tutti e tre i casi una protezione contro ingressi alti e per rientrare negli Absolute Maximum ratings nel caso di collegamento dei pin di output a massa (caso peggiore). (Vedi lo schematico delle parti: RASP\_GPIO\_DAC\_ADC\_PARTS.pdf)

#### 3.1.1 Filtri RC

Abbiamo disegnato due filtri RC:

- Filtro ADC.
- Filtro GPIO

Si tratta in entrambi i casi di filtri passa basso con 1 unico polo.

**Filtro ADC** L'ADC da specifiche si comporta come un passa basso a 5hz. Nel nostro caso vogliamo aggiungere un ulteriore polo in modo da **pre-filtrare** il segnale in ingresso. Scegliamo quindi una frequenza di taglio **<5Hz** (sopra alla quale in ogni caso il segnale non viene campionato correttamente). In particolare scegliamo una resistenza da 100k $\Omega$  e un condensatore da 1 $\mu$ F per avere di conseguenza un taglio a 1.6Hz.

**Filtro GPIO** Per il GPIO il caso è diverso: ci serve un filtro in modo da evitare false letture dovute a picchi improvvisi e rumore e nel contempo assicurare una buona reattività nel caso dell'attuazione. Dimensioniamo il filtro a partire dalla resistenza che deve soddisfare la massima corrente di drain dal piedino descritta negli **Absolute Maximum Ratings**. Abbiamo quindi  $I_{max}=25\text{mA}$ ,  $V_{max}=5\text{V}$ .  $R_{min}$  deve quindi essere  $R_{min}=200\Omega$ . La dimensioniamo ad  $1\text{k}\Omega$  e siamo sicuri di essere nei ratings. Scegliamo una  $C=1\mu\text{F}$  per avere di conseguenza un taglio a  $160\text{Hz}$ . La porta ha quindi una  $\tau=1\text{ms}$  di risposta naturale, che è accettabile nel nostro caso perché nel caso peggiore la commutazione del valore binario avviene intorno ai  $2\text{ms}$ .

### 3.1.2 Protezioni V e I

Ci serve un sistema per impedire alla tensione di salire oltre ad una certa soglia in modo da non danneggiare i microchip. utilizziamo quindi dei diodi zener che scaricano a massa le tensioni in ingresso. Si noti che le usiamo anche nel caso dei DAC per proteggere la porta nel caso in cui venga collegata erroneamente.

**Protezione DAC** Nel caso del DAC vogliamo proteggere la porta da tensioni troppo alte. Al massimo il DAC da  $V_{out}=V_{dd}=5\text{V}$ . Colleghiamo un diodo Zener con  $V_z=5.1\text{V}$  in inversa e qualunque tensione maggiore di  $5.1$  verrà scaricata dal componente. La resistenza viene scelta per rispettare la massima corrente erogata dal pin  $I_{max}=25\text{mA}$ . Con  $V=5\text{V}$  e  $I_{max}=25\text{mA}$ ,  $R_{min}=200\Omega$ . La dimensioniamo ad  $1\text{k}\Omega$ .

**Protezione GPIO** Pure nel caso del GPIO vogliamo proteggere la porta da tensioni troppo alte. Al massimo il GPIO da  $V_{out}=V_{dd}=5\text{V}$ . Colleghiamo un diodo Zener con  $V_z=5.1\text{V}$  in inversa e qualunque tensione maggiore di  $5.1$  verrà scaricata dal componente. La resistenza viene scelta per rispettare la massima corrente erogata dal pin  $I_{max}=25\text{mA}$ . Con  $V=5\text{V}$  e  $I_{max}=25\text{mA}$ ,  $R_{min}=200\Omega$ . La dimensioniamo ad  $1\text{k}\Omega$ .

**Protezione ADC** Nel caso dell'ADC vogliamo proteggere la porta da tensioni troppo alte. I max ratings sono a  $5\text{V}$ , ma qualunque segnale con tensione maggiore di  $V_{ref}$  non viene campionato correttamente, infatti abbiamo un fenomeno di clipping a  $V < -V_{ref}/PGA$  e a  $V > V_{ref}/PGA$ . (con  $PGA = 1,2,4,8$ ). L'intervallo utile su cui misurare il segnale sono quindi delle tensioni con la tensione  $-2.048\text{V} < V < 2.048\text{V}$ .

Colleghiamo un diodo Zener con  $V_z=2.2\text{V}$  in inversa e in modo da evitare questo compito al microchip. In questo caso la resistenza viene scelta per rispettare la massima corrente erogata, ma anche per scegliere condensatori più piccoli e quindi meno costosi. La dimensioniamo ad  $100\text{k}\Omega$ . Per usare un  $C=1\mu\text{F}$ . In ogni caso i segnali che leggeremo trasportano pochissima corrente, quindi possiamo filtrare la corrente massima molto pesantemente.

## 3.2 Schema elettrico d'assemblaggio