Documentazione parte locale

La parte locale del progetto SMASHBOX si compone di due rami principali: La parte di Acquisition e la parte Central.

La parte di Acquisition si compone di 4 sensori: 2 sensori ad infrarossi, un accelerometro e un sensore di temperatura e di umidità. Questi elementi saranno presenti in ogni cassetta.

La parte Central invece sarà unica e si compone di un rilevatore di impronte digitali e di una telecamera (forse?).

Acquisition

Lo scopo dei sensori presenti è quello di produrre un digital twin per ogni cassetta e per fornire una intercomunicazione tra le varie cassette.

In particolare lo scopo del primo sensore a infrarossi sarà quello di verificare la presenza o meno dell’oggetto all’interno della cassetta, il secondo sensore ad infrarossi invece avrà lo scopo di verificare se la cassetta è aperta o chiusa mentre il sensore di temperatura e l’accelerometro funzionano a lato security. Tramite questi sensori infatti sarà possibile monitorare se è in atto un tentativo di manomissione alla cassetta e nel caso mandare in stato di blocco anche tutte le altre.

La configurazione del microcontrollore e dei sensori è la seguente:

Immagine che contiene testo, Ingegneria elettronica, elettronica, circuito

Descrizione generata automaticamente

Iniziamo ora a descrivere i singoli sensori

**SENSORE DI TEMPERATURA E UMIDITA’: AHT25**

Caratteristiche:

* Protocollo di comunicazione I2C
* Precisione della temperatura ±0.3℃
* Precisione dell’umidità ±2%
* Alimentazione: 2.2-5.5V
* Risposta rapida e capacità anti-interferenza

Comunicazione:

il sensore sfrutta il protocollo di comunicazione I2C che funziona attraverso 4 wires:

* Vdd
* SDA (porta bidirezionale)
* GND
* SCL (porta bidirezionale)

Il serial clock (SCL) viene utilizzato per sincronizzare la comunicazione tra il microprocessore (nel nostro caso Arduino Uno) e il sensore AHT25. Siccome l’interfaccia contiene una logica completamente static non esiste una frequenza SCL minima da imporre. Il pin SDA invece viene utilizzato per l’input e l’output dei dati dal sensore. Quando viene inviato un comando al sensore SDA è valido sull’edge di salita del segnale SCL, mentre quando SCL è alto SDA rimane stabile. Dopo il fronte di discesa di SCL il valore di SDA può essere modificato.

Funzionamento:

Il primo passo è alimentare il sensore con la tensione VDD selezionata (intervallo compreso tra **2,2V e 5,5V**). Dopo l'accensione, il sensore necessita di meno di **100ms di tempo di stabilizzazione** (con **SCL alto** in questo periodo) per raggiungere lo stato idle ed essere pronto a ricevere comandi inviati dall'host. Ogni trasmissione inizia con lo stato di start e termina con lo stato di stop. Stato di start: Quando SCL è alto, SDA passa da alto a basso. Lo stato di start è uno stato speciale del bus controllato dal master, che indica l'inizio della trasmissione dello slave (dopo lo Start, il BUS è generalmente considerato in uno stato occupato). Stato di stop: Quando SCL è alto, la linea SDA passa da basso ad alto. Lo stato di arresto è uno stato speciale del bus controllato dal master, che indica la fine della trasmissione dello slave (dopo lo Stop, il BUS è generalmente considerato in uno stato inattivo). Dopo l'avvio della trasmissione, il primo byte trasmesso tramite I²C include: L'indirizzo del dispositivo I²C a 7 bit (0x38) e Un bit di direzione SDA (R per lettura: ‘1’, W per scrittura: ‘0’). Dopo l'ottava discesa del fronte del clock SCL, il pin SDA viene abbassato (bit ACK) per indicare che i dati del sensore sono stati ricevuti correttamente. Dopo l'invio del comando di inizializzazione (‘11100001’ per l'inizializzazione, ‘10101100’ per la misurazione di temperatura e umidità), l'MCU deve attendere fino al completamento della misurazione.

Processo di lettura del sensore: Dopo l'accensione, attendere almeno 100 ms. Prima di leggere i valori di temperatura e umidità, ottieni un byte della parola di stato inviando il comando 0x71. Se la parola di stato e 0x18 non sono uguali a 0x18, inizializzare i registri 0x1B, 0x1C, 0x1E. Se invece sono uguali, procedere al passaggio successivo. Attendere 10 ms e inviare il comando 0xAC per avviare la misurazione. Questo comando ha due parametri: il primo byte è 0x33 e il secondo byte è 0x00. Attendere 80 ms affinché la misurazione venga completata. Se il bit [7] della parola di stato letta è 0, significa che la misurazione è completata; a questo punto è possibile leggere sei byte consecutivi. Altrimenti, continuare ad attendere. Attendere 80 ms affinché la misurazione venga completata. Se il bit [7] della parola di stato letta è 0, significa che la misurazione è completata; a questo punto è possibile leggere sei byte consecutivi. Altrimenti, continuare ad attendere. Calcolare i valori di temperatura e umidità.

Calcolo temperatura e umidità: La formula per calcolare l'umidità relativa (RH) basandosi sul segnale SRH​ di umidità relativa, ottenuto tramite l'uscita SDA, è la seguente:

La temperatura si può ottenere sostituendo il segnale di output di temperatura St nella seguente formula:

Nel codice utilizzato questi passaggi sono stati fatti sfruttando le funzioni della libreria Adafruit\_AHTX0.h

**SENSORE DI DISTANZA AD INFRAROSSI: AZ-DELIVERY IR-ABSTAND SENSOR**

Caratteristiche:

* Alimentazione: 3.3-5.5V
* Contiene un potenziometro
* Sensibilità del sensore dipende dall’ambiente che lo circonda

Comunicazione:

3 pin:

* Output
* VCC
* GRD

Funzionamento:

La sensibilità del sensore IR viene regolata utilizzando il potenziometro. Il potenziometro può essere regolato in entrambe le direzioni. Per iniziare, regolare il potenziometro in senso orario fino a quando il LED indicatore inizia a illuminarsi. Una volta raggiunto questo punto, ruotare il potenziometro quanto necessario in senso antiorario per spegnere il LED indicatore. A questo punto, la sensibilità del ricevitore è massima e, di conseguenza, la distanza di rilevamento è massima. Se è necessario ridurre la distanza di rilevamento (cioè la sensibilità) del ricevitore, è possibile ruotare il potenziometro ulteriormente in senso antiorario a partire da questo punto. Inoltre, se l'orientamento dei LED Tx e Rx è parallelo la loro sensibilità è massima. Se vengono allontanati l'uno dall'altro, inclinando le estremità saldate l'una verso l'altra, la loro sensibilità si riduce. La sensibilità del sensore è limitata all'ambiente circostante. Una volta regolato per un determinato ambiente, funzioneranno perfettamente fino a quando le condizioni di illuminazione IR della zona rimarranno quasi costanti. Ad esempio, se il potenziometro viene regolato all'interno di una stanza o edificio per ottenere la massima sensibilità e successivamente il dispositivo viene portato all'aperto sotto la luce solare, sarà necessario regolarlo nuovamente, poiché i raggi solari contengono frequenze nell'infrarosso (IR) che agiscono come una fonte IR (trasmettitore). Questo interferisce con la capacità di rilevamento del ricevitore, rendendo necessaria una nuova regolazione per funzionare correttamente nel nuovo ambiente. L'uscita del ricevitore IR va a livello basso quando riceve un segnale IR. Pertanto, il pin di uscita è normalmente basso, poiché, anche se il LED IR trasmette continuamente, in assenza di ostacoli non viene riflesso nulla verso il ricevitore IR. In questa condizione, il LED indicatore rimane spento. Quando viene rilevato un ostacolo, l'uscita del ricevitore IR va a livello basso perché il segnale IR viene riflesso dalla superficie dell'ostacolo. Questo fa sì che l'uscita del comparatore vada a livello basso. Tale uscita è collegata al catodo del LED, che quindi si accende.

**ACCELEROMATRO: ADXL345**

Caratteristiche**:**

* Piccolo, sottile e a basso consumo accelerometro a 3 assi
* Alta risoluzione
* Misurazione fino a 16g
* Comunicazione SPI/I2C (nello specifico nel codice è stata utilizzara la I2C)
* Alimentazione: 2.0-3.6V

Comunicazione:

l’accelerometro ADXL345 può funzionare tramite due protocolli di comunicazione seriale: SPI e I2C. DI conseguenza presenta i pin necessari per entrambi i protocolli:

* GND
* VCC
* CS
* INT1
* INT2
* SDO
* SDA
* SCL

Siccome nella nostra implementazione è stato scelto il protocollo I2C i pin utilizzati saranno: gnd, vcc, sda e scl.

Con il pin CS alto a VDD I/O, l'ADXL345 opera in modalità I²C, richiedendo una semplice connessione a due fili, come mostrato nella figura corrispondente. Supporta le modalità di trasferimento dati standard (100 kHz) e veloce (400 kHz), a condizione che siano rispettati i parametri temporali indicati nelle tabelle e figure di riferimento. Sono supportate operazioni di lettura/scrittura su uno o più byte. Con il pin SDO/ALT ADDRESS impostato alto, l'indirizzo I²C a 7 bit del dispositivo è 0x1D, seguito dal bit di lettura/scrittura (R/W). Questo corrisponde a 0x3A per una scrittura e 0x3B per una lettura. Un indirizzo alternativo I²C, 0x53 (seguito dal bit R/W), può essere selezionato collegando a massa il pin SDO/ALT ADDRESS (Pin 12). Questo si traduce in 0xA6 per una scrittura e 0xA7 per una lettura.

Funzionamento:

L'ADXL345 è un sistema completo per la misurazione dell'accelerazione su tre assi, con un intervallo di misurazione selezionabile tra ±2 g, ±4 g, ±8 g o ±16 g. È in grado di misurare sia l'accelerazione dinamica, derivante da movimento o urti, sia l'accelerazione statica, come la gravità, il che ne consente l'uso come sensore di inclinazione. Il sensore è costituito da una struttura microlavorata in superficie con tecnologia al polisilicio, costruita sopra un wafer di silicio. Molle in polisilicio sospendono la struttura sopra la superficie del wafer e forniscono una resistenza alle forze di accelerazione. La deflessione della struttura viene misurata tramite condensatori differenziali composti da piastre fisse indipendenti e piastre collegate alla massa in movimento. L'accelerazione provoca la deflessione del fascio e squilibra il condensatore differenziale, generando un'uscita del sensore con un'ampiezza proporzionale all'accelerazione.

INVIO DATI AL BRIDGE

Tramite lo script svolto su Arduino IDE verranno inviati 8 byte al bridge tramite le porte seriali. Di seguito la legenda sul significato dei seguenti dati:

1: Presenza/assenza dell’oggetto nella cassetta. Se il Byte in questione sarà ad 1 l’oggetto sarà presente all’interno mentre 0 altrimenti. (Local -> Bridge)

2-3: Temperatura. Questi due byte vengono utilizzati per inviare la temperatura in gradi Celsius in formato decimale al bridge (Local -> Bridge)

4-5: Umidità: Questi due byte vengono utilizzati per inviare la percentuale di umidità nell’aria al bridge (Local -> Bridge)

6: infrazione/non infrazione: l’accelerometro rileva l’accelerazione del sensore lungo i 3 assi, qualora questa misura superi una certa soglia il byte viene impostato ad 1 altrimenti a 0. Ha scopo di rilevare se c’è un tentativo di manomissione fisica. (Local -> Bridge)

7: Lock/Unlock: byte che viene impostato ad 1 qualora vi sia stato un tentativo di manomissione su una qualsiasi delle cassette di sicurezza. La presenza di questo byte ad 1 implica il blocco all’apertura di tutte le cassette. La situazione può essere sbloccata solo da central (Local <- Bridge)

8: open/close: byte che indica se la cassetta è fisicamente aperta (1) o chiusa (0). (Local -> Bridge)

Central

La parte Central ha come scopo lo sblocco della corretta cassetta di sicurezza tramite un meccanismo a due fattori: l’inserimento della propria impronta digitale e l’indicazione del numero di cassetta attraverso un riconoscimento numerico. La lista delle impronte registrate e i numeri associati è contenuta sul cloud dunque lo scopo della parte central sarà quella di ottenere i dati che andranno confrontati con quelli contenuti nel server.

La parte di central si sviluppa in 2 componenti principali: un lettore di impronte digitali e una telecamera per il riconoscimento dei segni.

Immagine che contiene testo, cavo, presa, design

Descrizione generata automaticamente

**LETTORE DI IMPRONTE DIGITALI: JM-101**

Caratteristiche:

* alimentazione 3.3V
* comunicazione UART e USB

Pin:

* Vi
* Tx
* Rx
* GND
* Touch
* TouchVin
* D+ (USB)
* D- (USB)

Noi collegheremo solo i primi 4

Comunicazione:

UART: Prima che l'interfaccia UART trasmetta i pacchetti di dati, deve prima ricevere il pacchetto di istruzioni relativo ai dati da inviare, quindi inviare il pacchetto di richiesta una volta pronta la trasmissione. Infine, si procede con la trasmissione del pacchetto di dati. I pacchetti di dati includono principalmente: header del pacchetto, indirizzo del chip, identificatore del pacchetto, lunghezza del pacchetto, dati e checksum. L'identificatore del pacchetto è principalmente suddiviso in due tipi: 02H e 08H.  
02H: pacchetto di dati con pacchetti successivi mentre 08H: ultimo pacchetto, pacchetto finale. La lunghezza dei dati è preimpostata ed è generalmente suddivisa in: 32, 64, 128 e 256.

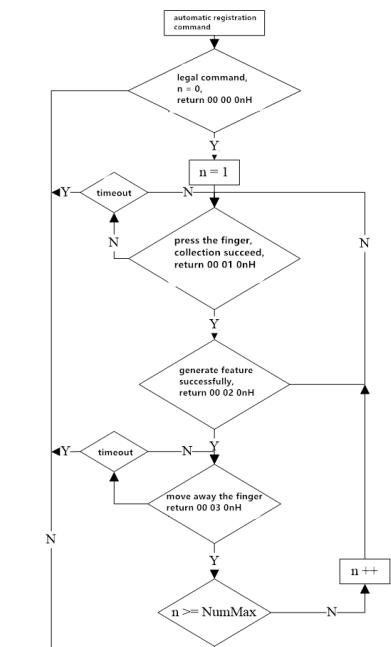
Ad esempio, se la lunghezza dei dati da trasmettere è di 1 Kbyte e la lunghezza dei dati per pacchetto è impostata a 128 byte, allora i dati da 1 Kbyte verranno suddivisi in 8 pacchetti. Ogni pacchetto comprende:

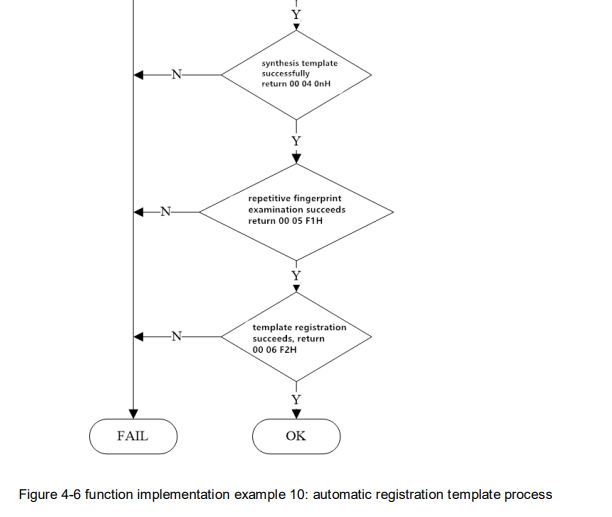
* 2 byte per l’header del pacchetto,
* 4 byte per l'indirizzo del chip,
* 1 byte per l'identificatore del pacchetto,
* 2 byte per la lunghezza del pacchetto,
* 128 byte per i dati,
* 2 byte per il checksum.

La lunghezza totale di ciascun pacchetto sarà quindi di 139 byte. Inoltre, negli 8 pacchetti: L'identificatore del pacchetto dei primi 7 è 02H e L'identificatore dell'ultimo pacchetto è 08H. Infine, è importante notare che, se la lunghezza di un pacchetto non raggiunge i 139 byte, non verrà estesa artificialmente a tale lunghezza in nessun altro modo.

USB: Prima della trasmissione USB, è necessario ricevere il pacchetto di istruzioni relativo al pacchetto di dati e trasmettere il pacchetto di richiesta di successo una volta che la trasmissione è pronta. I pacchetti di dati USB contengono solo i dati, senza intestazione del pacchetto, indirizzo del chip, identificatore del pacchetto, lunghezza del pacchetto o checksum. A differenza della trasmissione segmentata dell'interfaccia UART, la trasmissione USB avviene in pacchetti completi.

Processo di registrazione automatica di una nuova impronta digitale





Processo di verifica automatica dell’impronta digitale

