

Diari progetto WordClock

Diario di lavoro

Luogo	SAM Trevano
Data	16.03.2018

Lavori svolti

Oggi abbiamo dapprima fatto una riunione per ricevere i progetti, in seguito abbiamo ricevuto il QDC ed abbiamo cominciato ad occuparci della parte di analisi della documentazione. Abbiamo anche concluso il Gantt preventivo e l'abbiamo inserito nella documentazione. Inoltre abbiamo consultato il progetto del word-clock della vecchia classe di terza in modo da capire tutto ciò che dobbiamo modificare, aggiungere o sostituire.

Problemi riscontrati e soluzioni adottate

Nel capitolo che riguarda l'analisi non abbiamo fatto l'analisi dei costi, questo perché non sappiamo ancora tutti i materiali che necessitiamo.

Punto della situazione rispetto alla pianificazione

Siamo in orario rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Iniziare la parte di progettazione della documentazione.

Diario di lavoro

Luogo	SAM Trevano
Data	23.03.2018

Lavori svolti

Oggi abbiamo continuato con la documentazione visto che c'erano ancora lacune parti nel capitolo dell'analisi che dovevano essere sistemate. In seguito ci hanno portato il word-clock che è stato realizzato dai nostri compagni di quarta l'anno scorso e ci hanno spiegato che c'erano alcune cose che dobbiamo cambiare e/o sistemare come ad esempio tutto il circuito che è stato costruito nella parte "coperta" della struttura che costituiva la vecchia retroilluminazione è da smontare completamente questo perché al posto dei vecchi led dovrà essere montata una nuova striscia di led neopixel da 144 led al metro. In seguito ci siamo messi a progettare con fritzing lo schema elettrico che utilizzeremo per la gestione dei led, per l'accensione del word-clock (tramite bottone) e per la correzione dell'ora (tramite bottone e potenziometro). Inoltre abbiamo fatto uno schema elettrico con il funzionamento dei led Neopixel ed riorganizzato il Gantt siccome i prossimi 2 venerdì saranno feriali e di conseguenza non saremo a scuola.

Problemi riscontrati e soluzioni adottate

All'inizio della lezione quando ci siamo occupati della documentazione abbiamo avuto alcuni problemi con il "codice" in md ma alla fine abbiamo risolto il problema che era causato dal fatto che scrivendo la documentazione in md non si devono mettere indentature.

Punto della situazione rispetto alla pianificazione

Siamo in orario rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Iniziare la parte di progettazione della documentazione.

Diario di lavoro

Luogo	SAM Trevano
Data	13.04.2018

Lavori svolti

Oggi abbiamo dovuto cercare la densità di led giusta della striscia di neopixel da ordinare, per fare ciò abbiamo dovuto calcolare, in base alla grandezza di ogni quadrato che componeva la griglia, il “passo” ideale tra i led della striscia di neopixel (17mm di passo). In seguito Alessandro ha aggiunto la parte che permette di modificare l’ora al design del controllo onboard mentre Fabio sta testando il funzionamento del modulo RCT (Real Time Clock) che ci è stato dato dal sore, questo modulo dovrebbe potersi collegare e restare sempre connesso all’orario di Amburgo, dato che, funziona ad onde radio. Se dovessimo riuscire a farlo funzionare potremmo evitare di usare un modulo Wi-Fi che si collega a diversi time server a disposizione. Abbiamo rimosso la matrice di led che era montata sotto la struttura e abbiamo dissaldato tutti i 182 led.

Problemi riscontrati e soluzioni adottate

Abbiamo riscontrato alcuni problemi nel codice per il funzionamento dell’RTC.

Punto della situazione rispetto alla pianificazione

In orario sulla pianificazione.

Programma di massima per la prossima giornata di lavoro

Fare funzionare l’RTC.

Diario di lavoro

Luogo	SAM Trevano
Data	20.04.2018

Lavori svolti

Oggi ho pensato, grazie ad un suggerimento, un'idea per rendere più "semplice" il controllo OnBoard da parte dell'utente, quest'ultima consiste nel usufruire dei pallini posti in alto al foglio retroilluminato in modo tale che quando l'utente clicca sul bottone di reset per tre secondi si sta modificando le ore o i minuti, in base al pallino che si illuminerà. In seguito ho cercato di fare funzionare il modulo dcf77 ma ho riscontrato dei problemi con le versioni delle librerie. Una volta risolto questo problema il codice compilava e si eseguiva senza dare errori. Ora mi sono messo a cercare un modo per riuscire a collegare il DCF77 alla frequenza che trasmette l'orario corrente. Dopo varie ricerche ho letto che devo riuscire a collegarmi ad un segnale con frequenza 77.5 kHz, questo segnale viene trasmesso da Mainflingen, un paese vicino a Francoforte e questa trasmittente ha un raggio di 2000Km. Il problema è che non riesco a trovare nessun tipo di esempio di codice su internet che mi permetta di implementare tramite Arduino una connessione a questo segnale.

Problemi riscontrati e soluzioni adottate

Il codice preso dal sito di Arduino per il funzionamento del modulo dcf77 continuava dare errori durante la compilazione che riguardavano le due librerie che richiedeva per il suo corretto funzionamento (libreria Time e DCF77). L'errore era causato dal fatto che le librerie che erano presenti sul sito di Arduino avevano una versione molto vecchia (DCF77-0.9.8), però cercando su internet sono riuscito a trovare delle librerie più aggiornate (DCF77-1.0.0 e Time-1.5.0).

Punto della situazione rispetto alla pianificazione

In orario sulla pianificazione.

Programma di massima per la prossima giornata di lavoro

Riuscire a fare funzionare l'RTC e cominciare a montare i NeoPixel.

Diario di lavoro

Luogo	SAM Trevano
Data	27.04.2018

Lavori svolti

Oggi abbiamo continuato a lavorare sul modulo dcf77, dopo varie ricerche e grazie all'aiuto del sore abbiamo fatto un montaggio su breadboard per vedere se il dcf77 "manda in uscita" un segnale. Dopo vari tentativi continuavamo a notare che il segnale che dava in uscita nel codice in Arduino era sempre fisso e quindi errato. Quindi abbiamo preso l'antenna e l'abbiamo attaccata ad un oscilloscopio e al montaggio e, testandola, abbiamo constatato che l'antenna funziona ma il problema sta nel circuito integrato che, a causa di tutte le prove effettuate sia per capire la posizione dei pin sia per il segnale, probabilmente si è bruciato e non è più funzionante. In seguito abbiamo continuato con la documentazione aggiungendo un po' di capitoli e continuando con il capitolo della progettazione. In seguito abbiamo preso una striscia di led neopixel per cominciare a testare il suo funzionamento in modo tale che, appena arriverà quella che abbiamo ordinato, dovremmo solo montarla sulla struttura e adeguare il codice in base alle parole del wordclock. Per testarle ci siamo fatti passare il metodo che ha anche utilizzato il nostro compagno Igor per il suo progetto in modo da capirne il funzionamento base, per fare ciò abbiamo cercato anche le librerie che gestiscono i neopixel.

Problemi riscontrati e soluzioni adottate

L'antenna dcf77 funziona ma il suo circuito integrato non funziona, probabilmente si è bruciato e quindi il sore ce ne ha ordinata un'altra e anche un modulino rtc in caso non riuscissimo a fare funzionare l'antenna dcf77.

Punto della situazione rispetto alla pianificazione

In ritardo sulla pianificazione a causa della mancata consegna delle strisce di neopixel e anche a causa del circuito bruciato dell'antenna dcf77.

Programma di massima per la prossima giornata di lavoro

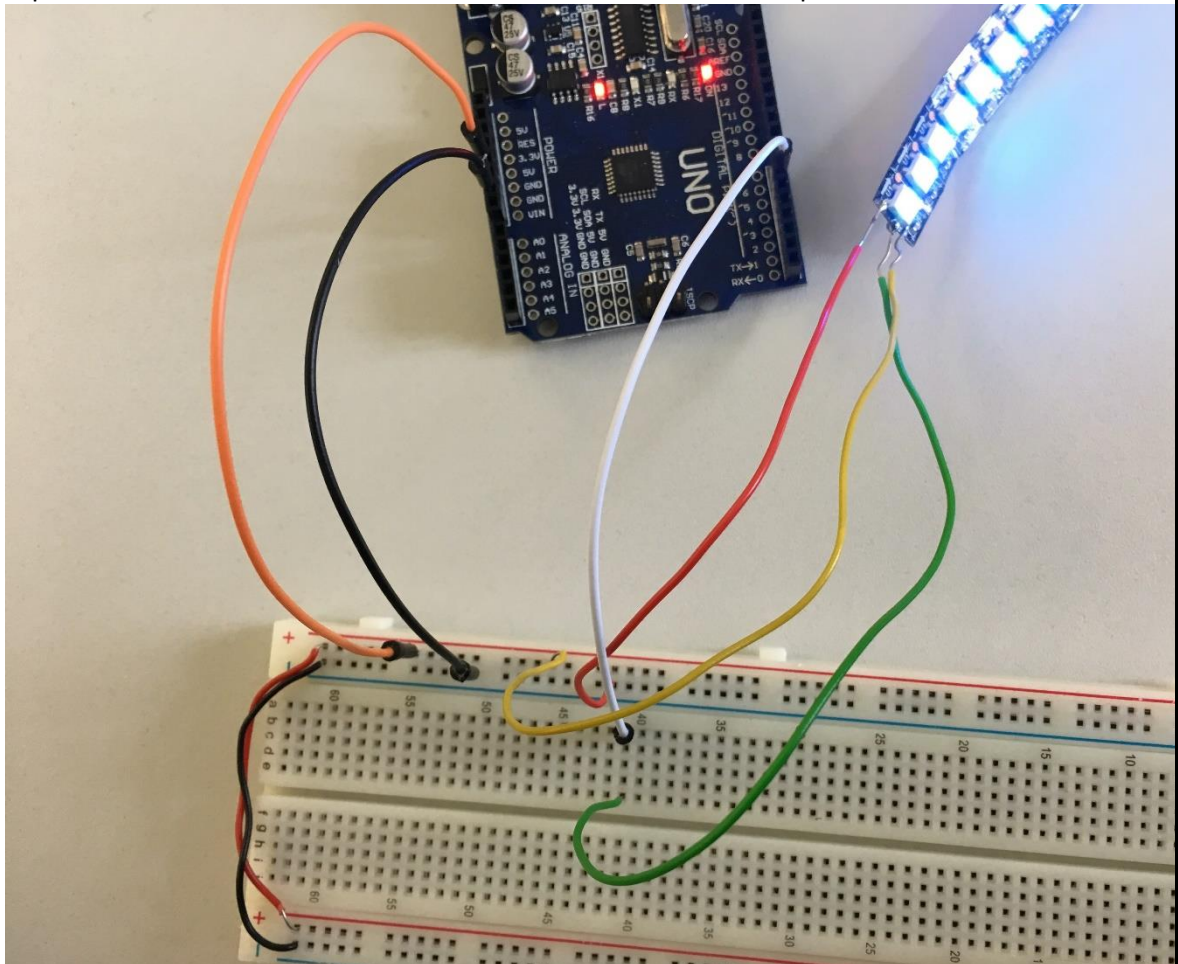
Riuscire a fare funzionare l'RTC e cominciare a montare i NeoPixel.

Diario di lavoro

Luogo	SAM Trevano
Data	4.05.2018

Lavori svolti

Oggi ci sono arrivate le due strisce di led neopixel e il modulo rtc. Innanzitutto abbiamo cominciato con il saldare 3 fili su una striscia di led neopixel di prova così da poter testare e di capire il suo funzionamento tramite anche il metodo che ci è stato passato.



In seguito abbiamo cominciato a tagliare le strisce di neopixel, in base alla griglia in legno del wordclock, le abbiamo tagliate con uno schema di: 3led, 4led, 3led, 3led. Una volta tagliate abbiamo preparato tutti i cavi che ci serviranno per collegare tutte le strisce tagliate tra loro. Una volta preparato anche tutti i cavi abbiamo cominciato a saldare le strisce. Mentre Fabio si occupava di saldare Alessandro si occupava del codice che utilizzeremo per fare illuminare tutti i neopixel in base alla parola del wordclock che si deve illuminare. Non abbiamo ancora redatto nessun codice perché prima dobbiamo pensare ad un metodo per gestire l'accensione delle parole in base all'orario corrente. Inizialmente abbiamo pensato di utilizzare una matrice ma quest'idea l'abbiamo scartata subito per via di come abbiamo deciso di montare i neopixel (in verticale).

Problemi riscontrati e soluzioni adottate

Un problema che abbiamo riscontrato è quello di trovare un modo, che non includa una lista di if, per implementare in maniera elegante e funzionale l'accensione dei led neopixel in base all'orario corrente.

Punto della situazione rispetto alla pianificazione

In ritardo sulla pianificazione.

Programma di massima per la prossima giornata di lavoro

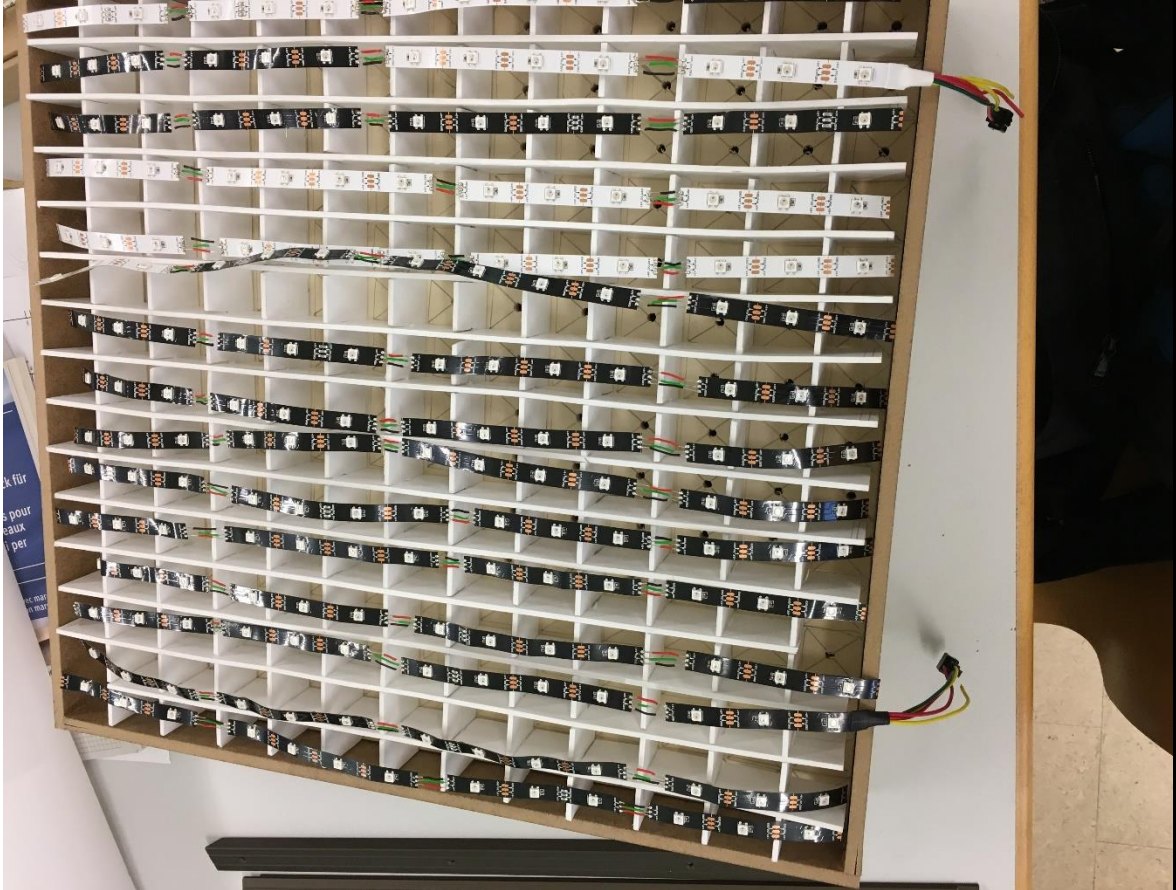
Finire di montare tutta la struttura e portare avanti il codice per la gestione del wordclock.

Diario di lavoro

Luogo	SAM Trevano
Data	11.05.2018

Lavori svolti

Oggi Fabio si è occupato di continuare e di finire di saldare tutte e 15 le strisce di led.



Mentre Alessandro si è occupato del codice per gestire le varie accensioni dei led in base alle parole. In primo luogo ha scritto un foglio contenente tutte le parole che dovranno essere illuminate (ore, minuti, secondi e altre) con la loro rispettiva riga:

Wired Clock

Ore:

UNA OVA (1.00/2.00)	1.00
MEZZOGIORNO (12.00)	12.00
DUE (2.00/4.00)	2.00
TRE (3.00/4.00)	3.00
QUATTRO (4.00/6.00)	4.00
SEI (6.00/8.00)	6.00
CINQUE (5.00/12.00)	5.00
SETTE (7.00/8.00)	7.00
OTTO (8.00/10.00)	8.00
NOVE (9.00/11.00)	9.00
DIECI (10.00/12.00)	10.00
UNDICI (11.00/12.00)	11.00
MEZZANOTTE (0.00)	0.00

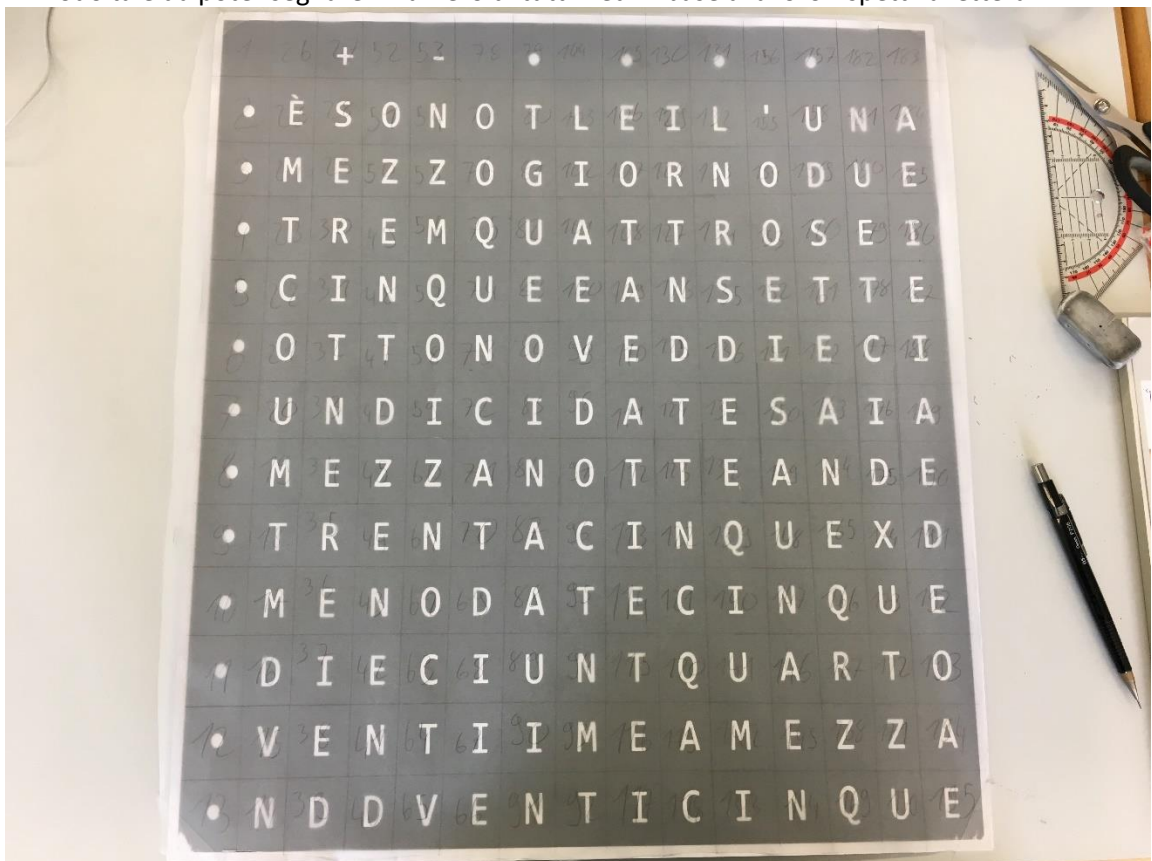
MINUTI: DUEX

TRENTACINQUE	35
GRACIA TRENTA	30
GRACIA	25
CINQUE	5
DIECI	10
VENTI	20
VENTICINQUE	25

ALTRO:

SONO	1
LE	2
LE	3
LE	4
LE	5
LE	6
LE	7
LE	8
LE	9
LE	10
LE	11
LE	12

Una volta fatto ciò gli è stato dato un foglio "trasparente" su cui ha potuto disegnare una griglia, in modo tale da poter segnare il numero di tutti i led in base alla loro rispettiva lettera.



1	26	27	52	53	78	79	104	105	130	131	156	157	182	183
2	25	28	51	54	77	80	103	106	129	132	155	158	181	184
3	24	29	50	55	76	81	102	107	128	133	154	159	180	185
4	23	30	49	56	75	82	101	108	127	134	153	160	179	186
5	22	31	48	57	74	83	100	109	126	135	152	161	178	187
6	21	32	47	58	73	84	99	110	115	136	151	162	177	188
7	20	33	46	59	72	85	98	111	114	137	150	163	176	189
8	19	34	45	60	71	86	97	112	113	138	149	164	175	190
9	18	35	44	61	70	87	96	113	122	139	148	165	174	191
10	17	36	43	62	69	88	95	114	121	140	147	166	173	192
11	16	37	42	63	68	89	94	115	120	141	146	167	172	193
12	15	38	41	64	67	90	93	116	119	142	145	168	171	194
13	14	39	40	65	66	91	92	117	118	143	144	169	170	195

Una volta fatto ciò si è rimesso a redigere il codice, innanzitutto ha creato tutti gli array delle varie parole, lettere e simboli con il numero dei rispettivi led.

```

//Array per gestire i "casi speciali":
int colonnaSecondi[13]; //Array contenente il numero dei led corrispondenti alla riga verticale dei pallini che segnano i secondi con multipli di 5..
int rigaPalliniMinuti[4] = {79, 105, 131, 157}; //Array contenente il numero dei led corrispondenti alla riga orizzontale dei pallini che segnano i minuti tra 1 e 4.
int piuMeno[2] = {27, 53}; //Array contenente il numero dei led corrispondenti al + ed al - posti nella riga orizzontale in cima.
int e[1] = {190}; //Array contenente il numero dei led corrispondente alla lettera: "E".
int meno[4] = {17, 36, 43, 62}; //Array contenente il numero dei led corrispondente alla parola: "meno";

//Array delle ore:
int eUna[6] = {25, 132, 155, 158, 181, 184}; //Array contenente il numero dei led corrispondenti alla scritta: "È l'una".
int sonoLeDue[9] = {28, 51, 54, 77, 103, 106, 159, 180, 185}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le due".
int sonoLeTre[9] = {28, 51, 54, 77, 103, 106, 23, 30, 49}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le tre".
int sonoLeQuattro[13] = {28, 51, 54, 77, 103, 106, 75, 82, 101, 108, 127, 134, 153}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le quattro".
int sonoLeCinque[12] = {28, 51, 54, 77, 103, 106, 22, 31, 48, 57, 74, 83}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le cinque".
int sonoLeSei[9] = {28, 51, 54, 77, 103, 106, 160, 179, 186}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le sei".
int sonoLeSette[11] = {28, 51, 54, 77, 103, 106, 135, 152, 161, 178, 187}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le sette".
int sonoLeOtto[10] = {28, 51, 54, 77, 103, 106, 21, 32, 47, 58}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le otto".
int sonoLeNove[10] = {28, 51, 54, 77, 103, 106, 73, 84, 99, 110}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le nove".
int sonoLeDieci[11] = {28, 51, 54, 77, 103, 106, 136, 151, 162, 177, 188}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le dieci".
int sonoLeUndici[12] = {28, 51, 54, 77, 103, 106, 20, 33, 46, 59, 72, 85}; //Array contenente il numero dei led corrispondenti alla scritta: "Sono le undici".
int eMezzogiorno[12] = {25, 24, 29, 50, 55, 76, 81, 102, 107, 128, 133, 154}; //Array contenente il numero dei led corrispondenti alla scritta: "È mezzogiorno".
int eMezzanotte[11] = {25, 19, 34, 45, 60, 71, 86, 97, 112, 123, 138}; //Array contenente il numero dei led corrispondenti alla scritta: "È mezzanotte".

//Array dei minuti:
int cinque[6] = {121, 140, 147, 166, 173, 192}; //Array contenente il numero dei led corrispondenti alla parola: "Cinque".
int dieci[5] = {16, 37, 42, 63, 58}; //Array contenente il numero dei led corrispondenti alla parola: "Dieci".
int unQuarto[8] = {89, 94, 120, 141, 146, 167, 172, 193}; //Array contenente il numero dei led corrispondenti alla scritta: "Un quarto".
int venti[5] = {15, 38, 41, 64, 67}; //Array contenente il numero dei led corrispondenti alla parola: "Venti".
int venticinque[11] = {65, 66, 91, 92, 117, 118, 143, 144, 169, 170, 195}; //Array contenente il numero dei led corrispondenti alla parola: "Venticinque".
int trenta[6] = {18, 35, 44, 61, 70, 87}; //Array contenente il numero dei led corrispondenti alla parola: "Trenta".
int trentacinque[12] = {18, 35, 44, 61, 70, 87, 96, 113, 122, 139, 148, 165}; //Array contenente il numero dei led corrispondenti alla parola: "Trentacinque".

```

In seguito ha creato 4 metodi: uno per gestire il cambio dell'ora, uno per gestire l'accensione dei simboli + e -, uno per gestire l'illuminazione dei led che rappresentano i minuti da 1 a 4 mentre l'ultimo serve per gestire l'illuminazione dei secondi.

```

//Metodo che gestisce le ore.
//Se i minuti superano i 40 allora incrementa l'ora di 1 e si attiva la parola meno.
void setOra() {
    if (minuti >= 40) {
        ora = ora + 1;
        meno = true;
    }
}

```

```

//Metodo che gestisce i simboli + e -.
//Se la scritta meno non è attiva allora accendo il led corrispondente al simbolo +
//Altrimenti accendo il led corrispondente al -.
void setPiuMeno() {
    if (meno != true) {
        strip.setPixelColor(piuMeno[0], 255, 0, 255);
        setMinutiPallini();
    } else {
        strip.setPixelColor(piuMeno[1], 255, 0, 255);
        setMinutiPallini();
    }
    strip.show();
}

```

```

//Metodo che gestisce i minuti tra l'1 ed il 4 segnalati con i pallini.
//Ad ogni ciclo accendo un led ed incremento i minuti.
void setMinutiPallini() {
    for (int i = 0; i < sizeof(rigaPalliniMinuti); i++) {
        strip.setPixelColor(rigaPalliniMinuti[i], 255, 0, 255);
        minuti = minuti + 1;
    }
}

```

```
//Metodo che gestisce la riga verticale dei secondi rappresentati con dei pallini.  
//Se quel determinato secondo é un multiplo di 5 accendo il suo rispettivo led,  
//quando arrivo a 60 secondi spengo tutti i led.
```

```
void setSecondi() {  
    int secondiP = secondi % 5;  
    int counter = 0;  
    if (secondiP == 0) {  
        switch (secondi) {  
            case 5:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 10:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 15:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 20:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 25:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 30:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 35:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 40:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 45:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 50:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 55:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            case 60:  
                strip.setPixelColor(colonnaSecondi[counter], 255, 0, 255);  
                counter++;  
                break;  
            default:  
                for (int i = counter; i <= 0; i--) {  
                    strip.setPixelColor(colonnaSecondi[i], 0, 0, 0);  
                }  
                counter = 0;  
                break;  
        }  
    }  
}
```

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione
In ritardo sulla pianificazione.

Programma di massima per la prossima giornata di lavoro
Finire di montare tutta la struttura e portare avanti il codice per la gestione del wordclock.

Diario di lavoro

Luogo	SAM Trevano
Data	18.05.2018

Lavori svolti

Oggi Fabio ha saldato tutte le strisce di led neopixel assieme e, in seguito, le ha attaccate alla struttura.



In seguito ha praticato un foro sulla struttura interna in modo tale da poterci fare passare i cavi che fungeranno da collegamento tra le strisce di led ed Arduino.



Mentre Alessandro si è occupato di portare avanti il codice per gestire l'illuminazione delle parole per il word clock. Al codice già esistente ho aggiunto due metodi: uno che si occupa di accendere tutti i led di ogni array (array passato come parametro), mentre l'altro metodo si occupa di spegnere tutti i led della striscia.

```
//Metodo che gestisce l'accensione dei led di ogni array.
```

```
void ledsOn(int scritta[], int grandezza) {
    for (int i = 0; i < grandezza; i++) {
        strip.setPixelColor(scritta[i], r, g, b);
        strip.show();
        Serial.print("ledON:");
        Serial.println(scritta[i]);
    }
}
```

```
//Metodo che spegne tutti i led della striscia.
```

```
void spegniStripeLed() {
    for (int i = 0; i < stripeLength; i++) {
        strip.setPixelColor(stripeLength[i], 0, 0, 0);
        strip.show();
        Serial.print("ledOff");
    }
}
```

In seguito ho apportato delle modifiche al metodo già esistente setOra() (funzionamento scritto sul file di testo posto nella stessa cartella del codice Arduino → *Gestione illuminazione ore e minuti.txt*, ma codice non concluso):

```
//Metodo che gestisce le ore.
```

```
//Se i minuti superano i 40 allora incrementa l'ora di 1 e si attiva la parola meno.
```

```
void setOra() {
    Serial.println("Gestione ora");
    if (minuti >= 40) {
        ora = ora + 1;
        Serial.print("Ora:");
        Serial.println(ora);
        m = true;
        minutiDaSottrarre = 60 - minuti;
        gestioneMinuti = minutiDaSottrarre % 5;
        if (gestioneMinuti == 0) {
            switch (minutiDaSottrarre) {
                case 5:
                    Serial.print("Minuti da sottrarre: 5");
                    ledsOn(cinque, sizeof(cinque));
                    break;
                case 10:
                    Serial.print("Minuti da sottrarre: 10");
                    ledsOn(dieci, sizeof(dieci));
            }
        }
    }
}
```

```

        break;
    case 15:
        Serial.print("Minuti da sottrarre: 15");
        ledsOn(unQuarto, sizeof(unQuarto));
        break;
    case 20:
        Serial.print("Minuti da sottrarre: 20");
        ledsOn(venti, sizeof(venti));
        break;
    }
    }else{
    }
}
}
}

```

Ha eseguito una modifica anche per quanto riguarda il metodo che gestisce l'accensione dei pallini che rappresentano i secondi (setSecondi()), ho tolto lo switch e ho "compattato il tutto" tramite un ciclo:

//Metodo che gestisce la riga verticale dei secondi rappresentati con dei pallini.
 //Se quel determinato secondo é un multiplo di 5 accendo il suo rispettivo led,
 //quando arrivo a 60 secondi spengo tutti i led.

```

void setSecondi() {
    for (int i = 0; i < (int)(secondi / 5); i++) {
        strip.setPixelColor(colonnaSecondi[i], r, g, b);
        Serial.print("Secondi: ");
        Serial.println(secondi);
        strip.show();
    }
}

```

Come ultima cosa ha creato nel loop() uno switch con 12 case che rappresentano le ore:

```

void loop() {
    setOra();
    Serial.print("Inizio");
    switch (ora) {
        case 0:
            Serial.print("Ora: 0");
            ledsOn(eMezzanotte, sizeof(eMezzanotte));
            break;
        case 1:
            Serial.print("Ora: 1");
            ledsOn(eLUna, sizeof(eLUna));
            break;
        case 2:
            Serial.print("Ora: 2");
            ledsOn(sonoLeDue, sizeof(sonoLeDue));
            break;
        case 3:
            Serial.print("Ora: 3");
            ledsOn(sonoLeTre, sizeof(sonoLeTre));
            break;
        case 4:
            Serial.print("Ora: 4");
            ledsOn(sonoLeQuattro, sizeof(sonoLeQuattro));
            break;
        case 5:
            Serial.print("Ora: 5");
            ledsOn(sonoLeCinque, sizeof(sonoLeCinque));
            break;
        case 6:
            Serial.print("Ora: 6");
            ledsOn(sonoLeSei, sizeof(sonoLeSei));
            break;
        case 7:
            Serial.print("Ora: 7");
            ledsOn(sonoLeSette, sizeof(sonoLeSette));
            break;
        case 8:
            Serial.print("Ora: 8");
            ledsOn(sonoLeOtto, sizeof(sonoLeOtto));

```



```

        break;
    case 9:
        Serial.print("Ora: 9");
        ledsOn(sonoLeNove, sizeof(sonoLeNove));
        break;
    case 10:
        Serial.print("Ora: 10");
        ledsOn(sonoLeDieci, sizeof(sonoLeDieci));
        break;
    case 11:
        Serial.print("Ora: 11");
        ledsOn(sonoLeUndici, sizeof(sonoLeUndici));
        break;
    case 12:
        Serial.print("Ora: 12");
        ledsOn(eMezzogiorno, sizeof(eMezzogiorno));
        break;
    default:
        Serial.print("Led spenti:");
        spegniStripeLed();
        break;
}

```

Come ultima cosa abbiamo si è occupato della presentazione.

Problemi riscontrati e soluzioni adottate

Il codice non è stato concluso e riscontrava alcuni problemi dopo la compilazione. Quando abbiamo eseguito il test sulle strisce di led montate sulla struttura si illuminavano solo le prime sei colonne, questo problema era causato da un filo che provocava la connessione tra il pin del dato e quello della massa, abbiamo risolto semplicemente tagliando il filo.

Punto della situazione rispetto alla pianificazione

In ritardo sulla pianificazione.

Programma di massima per la prossima giornata di lavoro

Presentazione del progetto.