

Część softwareowa naszego projektu w rozumieniu operacji wykonywanych na serwerze ujęta została do analizy obrazu – kodu podobnego do kodu QR, a następnie na podstawie jego analizy do wysłania informacji zwrotnej do osoby wykonującej zamówienie.

Zaczynając od samego początku, użytkownikowi na okularze wyświetlony zostanie kod, składający się z 4 części tj:

A###B###C###D###

gdzie za ### można wpisać dowolną liczbę nie większą niż 256. Każda z powyższych pozycji kryjąca się za poszczególną literą to unikatowa informacja dotycząca konkretnej lokacji na magazynie, do której magazynier musi się udać w pierwszej kolejności. W przykładowym rozumieniu może to być:

A – Numer korytarza

B – Lokacja wewnątrz korytarzu

C – Specjalny znak1

D – Specjalny znak2

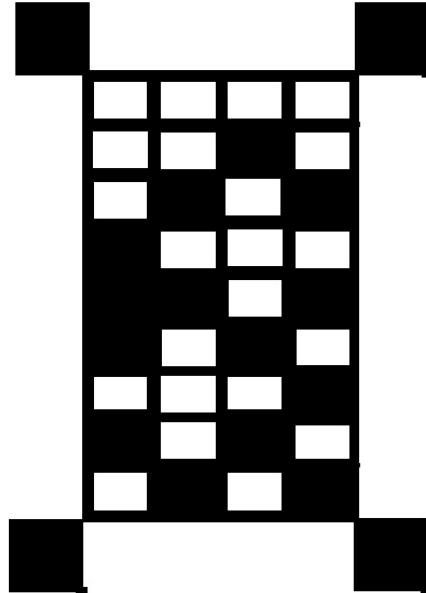
Za specjalne znaki można np. ustawić typ hali magazynowej (dla magazynu spożywczego może być to podział na chłodnię, mroźnię i magazyn o normalnej temperaturze), czy sytuacje, w których na jednej lokacji produktu znajdują się towary jednego typu ale o różnych cechach wyjątkowych np. batoniki „Princessy” o różnych smakach.

Przykładowo więc, niech będzie to produkt będący w korytarzu nr 22, na pozycji nr 14, znajdujący się na chłodni, o kodzie 111, i produkt ten jest produktem standardowym o specjalnym znaku2 równym 1. Kod takiego produktu wynosić więc będzie:

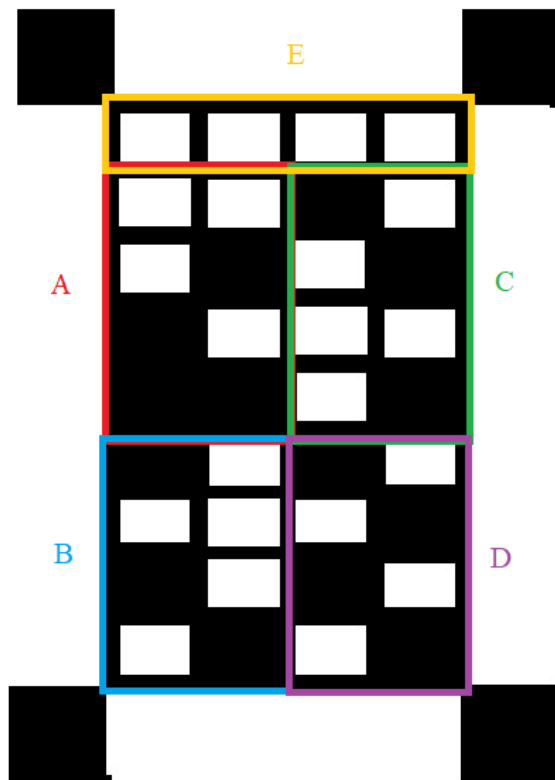
A22B14C111D1

Do takiej lokacji, więc pracownik magazynu ma obowiązek się udać po produkt bądź produkty.

Będąc już na miejscu, w celu uniknięcia pomyłki z dotarciem do nieprawidłowej lokalizacji, umieszczony będzie przy każdym miejscu występowania danego produktu kod, który pracownik ma obowiązek czytać za pomocą kamery znajdującej się w okularach. Kod ten jest unikatowy dla każdego miejsca poboru danego produktu. Prezentuje się on następująco:



Kod ten można podzielić na kilka sekcji, które są wykorzystywane przez program.



Najwyższa z sekcji – sekcja E, odpowiada za informację dla programu, że w tym miejscu kod się rozpoczyna. Nie jest to jej jedynym celem. Początkowe blozki informują o długości i wysokości wszystkich blozków poniższych w całym kodzie, gdyż należy pamiętać, że kod nie zawsze będzie czytany przez pracownika prostopadle do osi kamery, a pod pewnym kątem – czyli w pewnej perspektywie, która do celów poprawnego działania oprogramowania musi zostać na początku ustalona.

Sekcja A, jest to pierwsza z sekcji lokalizacji, która jest odpowiedzialna za numer korytarzu, w którym znajduje się produkt. Jest to w rozumieniu matematycznym macierz o 2 kolumnach i 4 wierszach, które są tak naprawdę kodem binarnym o budowie:



2^0	2^1
2^2	2^3
2^4	2^5
2^6	2^7

Białe pola w tej macierzy oznaczają binarne jedynki, natomiast czarne, binarne zera. Macierz powyższa ma wartość:

$$2^0 + 2^1 + 2^2 + 2^5 = 39$$

jak więc widać, sekcja A dla tego kodu przyjmuje wartość 39.

Pozostałe z sekcji mają się identycznie co sekcja A, a ich litery oznaczają to samo co zostało opisane we wcześniejszej części dokumentu.

Tak więc po analizie przedstawionego kodu, adres do którego ma udać się magazynier to:

A39B110C118D102

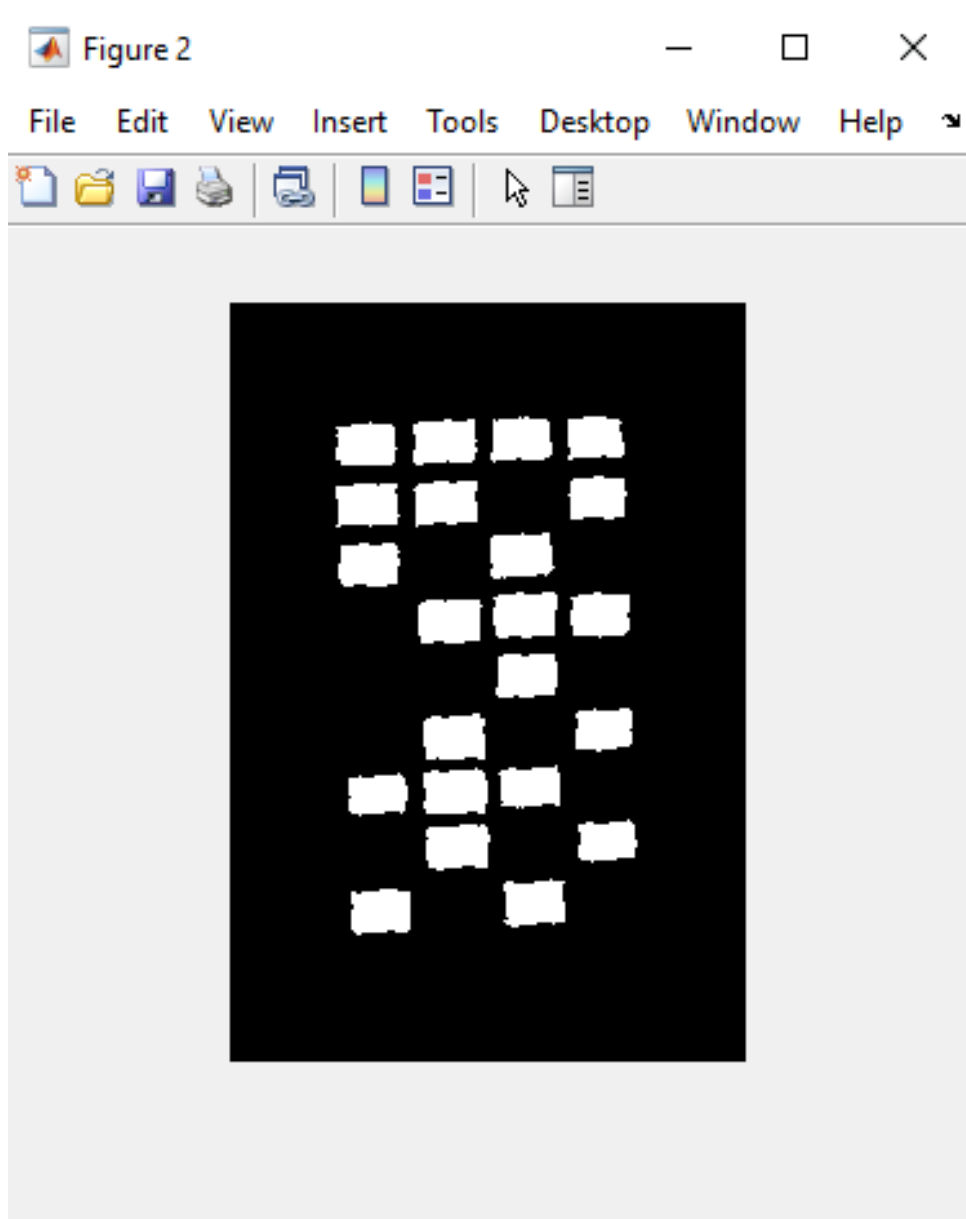
Gdy sczytana lokalizacja będzie zgodna z tym co zostało wyświetlone na okularze magazyniera przed sczytaniem, oprogramowanie wysyła zgodę, na pobór produktu, co skutkować będzie wyświetleniem ilości towaru jaki magazynier ma zapakować na kontener bądź paletę, czego obraz przedstawiony został poniżej:



Co do samego procesu analizy kodu, to prezentuje się on następująco:

1. Rozpoczęcie wysyłania obrazu rozpoczyna się w momencie, gdy akcelerometr będzie wysyłał wartości przyspieszeń bliskie zeru, co świadczyć ma o tym, że magazynier stoi.
2. Sczytany obraz poddany jest procesowi zamiany go na skale szarości.
3. Na podstawie adaptacyjnego progu binaryzacji, wyszukiwany jest obiekt w postaci całego kodu o konkretnych cechach dotyczących wypełnienia obiektu oraz jego zaokrąglenia.
4. Uzyskany kod w sytuacji poboru obróconego obrazu z kodem w lewo bądź prawo, doprowadzany jest do postaci, w której krawędzie kodu są równoległe do krawędzi obrazu
5. Mając obrócony kod, którego głównymi wyznacznikami pozycji są 4 kwadraty znajdujące się w jego rogach, wycinany jest z całego obrazu kamery jedynie obszar tzw. BoundingBox, zawierający w sobie obszar o krawędziach wspólnych z krawędziami zewnętrznymi kwadratów pozycjonujących powiększony o delikatny dystans, który jest konieczny w dalszym procesie obróbki kodu.

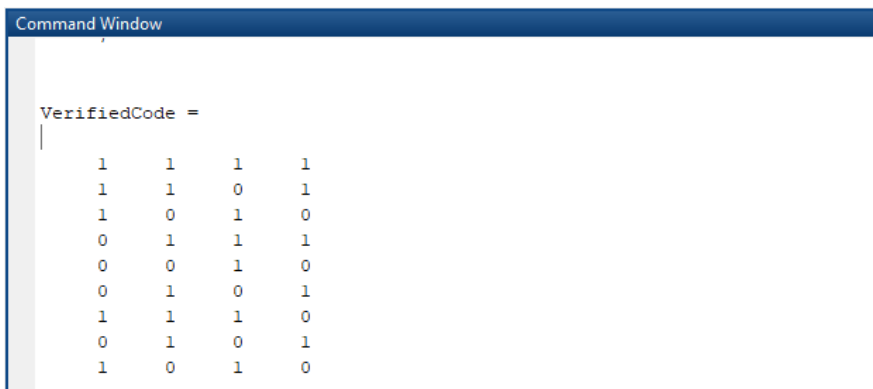
6. Znając stałe cechy obrazu pól w kolorze białym, wyznaczone zostały warunki determinujące obraz zawierający jedynie białe pola. Obraz ten prezentuje się następująco:



Jak więc można zauważyć, w tym momencie już otrzymany został obraz, identyczny jak ten z przykładu opisanego kilka stron wcześniej. Kod ten poddany zostaje dalszej analizie.

7. Na podstawie pierwszych czterech białych pól w pierwszym wierszu całego kodu, wyznaczana jest ilość pól w danej kolumnie iterując po każdej z nich. Korzystamy tutaj z centroidów danego obiektu (białego pola) i ich odległości od początkowego determinującego wysokość i szerokość pola startowego.

8. Finalnym efektem przetwarzania obrazu jest macierz zero-jedynkowa, która idealnie odwzorowuje kod z kamery. Przykład wyniku z konsoli programu Matlab pokazany został poniżej:



```
Command Window

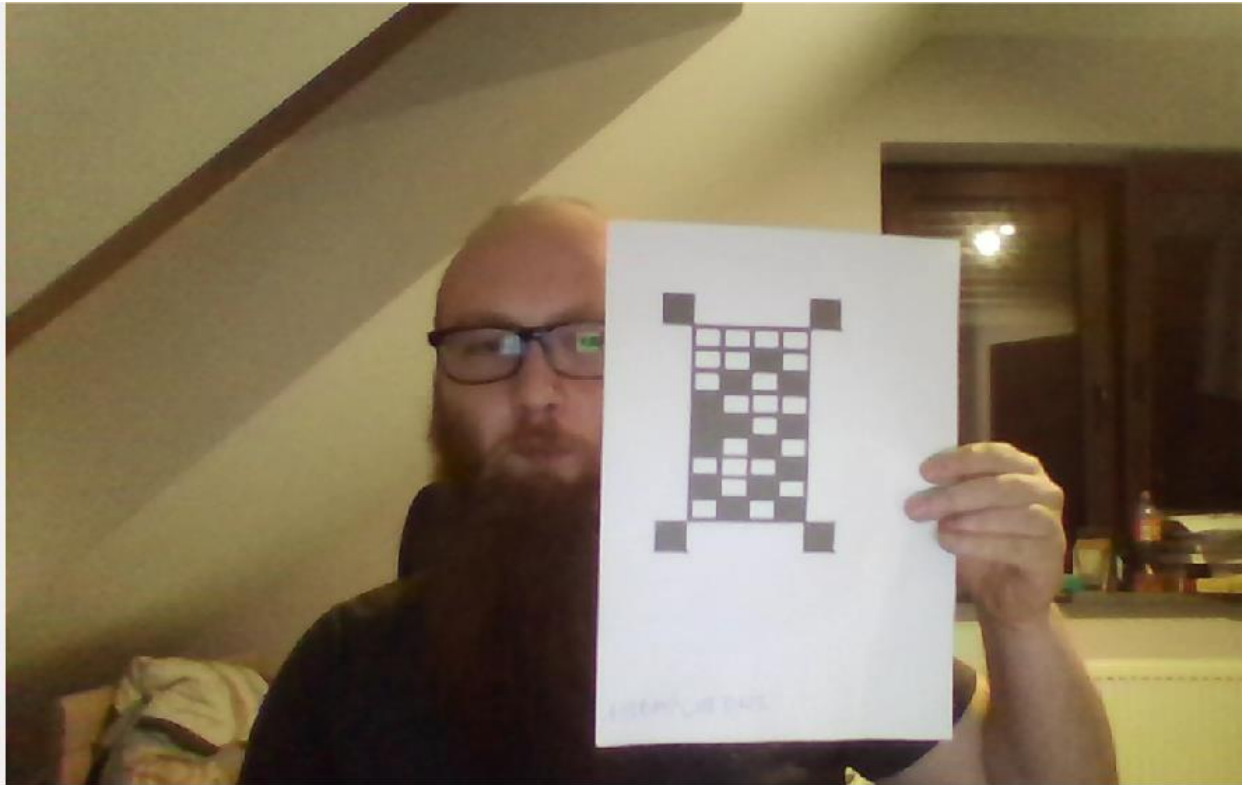
VerifiedCode =

     1     1     1     1
     1     1     0     1
     1     0     1     0
     0     1     1     1
     0     0     1     0
     0     1     0     1
     1     1     1     0
     0     1     0     1
     1     0     1     0
```

9. Powyższe operacje wykonywane są 7 razy i jeśli w 7 próbach kod się nie zmienia to jest on uznany za poprawny i przekazany do porównania z referencyjną lokalizacją. Taki zabieg został zastosowany, gdyż istnieje ryzyko, że z racji złego światła, bądź innych zakłóceń zewnętrznych kod przy jednokrotnej próbie może zostać sczytany niepoprawnie.
10. W sytuacji niepoprawnego sczytania kodu 7 krotnie, serwer wysyła informację do sterownika by ten poruszył serwomechanizmem w dół o jeden stopień, aż do pozycji minimalnej, a następnie w górę, aż do pozycji maksymalnej. W każdej iteracji obrotu serwomechanizmu, pobieranych jest znowu 7 wyników. Taki zabieg zastosowany został w sytuacji, gdy środek kodu znajduje się znacznie niżej bądź znacznie wyżej niż środek obrazu kamery i chcemy tą różnicę doprowadzić do jak najmniejszej wartości.
11. Kod binarny jest następnie przerabiany na kod w postaci liter ABCD oraz cyfr i porównywany z lokacją referencyjną, która została wyświetlona na okularze.
12. Gdy kod lokacji z programu będzie zgodny z lokacją referencyjną na okularach ukazywana jest ilość produktu, która magazynier musi pobrać.
13. Po pobraniu towaru, magazynier wciska przycisk znajdujący się na pudełku okularów informując system, że ta ilość produktu została wzięta i można przejść do kolejnej lokacji. W sytuacji braku towaru, użytkownik musi zatwierdzić swoją pozycję w postaci sczytania kodu z danej lokacji, jako informacja, że jest na pewno na lokacji wymaganej przez system, a potem za pomocą przycisku drugiego ten brak towaru zatwierdzić.

Wykonany został również test szczytywania kodów. Polegał on na wydrukowaniu różnych randomowych kodów, a następnie sprawdzenia czasów w jakich programowi udało się otrzymać wynikowy adres. Cały proces wykonywania testu opisany został za pomocą grafik ilustrujących ekran monitora:

1. Ustawienie kodu w widocznym dla kamery polu:



2. Uzyskanie wskaźnika prób g o wartości równej 7

```
g =  
  1  
  
g =  
  2  
  
g =  
  3  
  
g =  
  4  
  
g =  
  5  
  
g =  
  6  
  
g =  
  7
```

3. Otrzymanie kodu w postaci binarnej:

```
VerifiedCode =  
  
1      1      1      1  
1      1      0      1  
1      0      1      0  
0      1      1      1  
0      0      1      0  
0      1      0      1  
1      1      1      0  
0      1      0      1  
1      0      1      0
```

4. Zapisanie czasu potrzebnego na uzyskanie powyższego kodu wraz ze sprawdzeniem lokacji z tą referencyjną:

```
time =  
  
3.7607  
  
location =  
  
"A39B110C118D102"
```

Przeprowadzone na żywo 2 testy, załączone zostały w postaci filmików. Testy te zostały wykonane dla lokacji referencyjnej:

A11B27C52D108

Pierwszy z nich o nazwie Test_Wrong przedstawia sytuację z kodem będącym z innej lokacji, by pokazać, że magazynier jest sprawdzany w poprawny sposób.

Drugi z nich o nazwie Test_Correct pokazuje sytuację, w której pracownik znajduje się w dobrej lokalizacji, a czytany kod jest kodem zgodnym z referencją.

