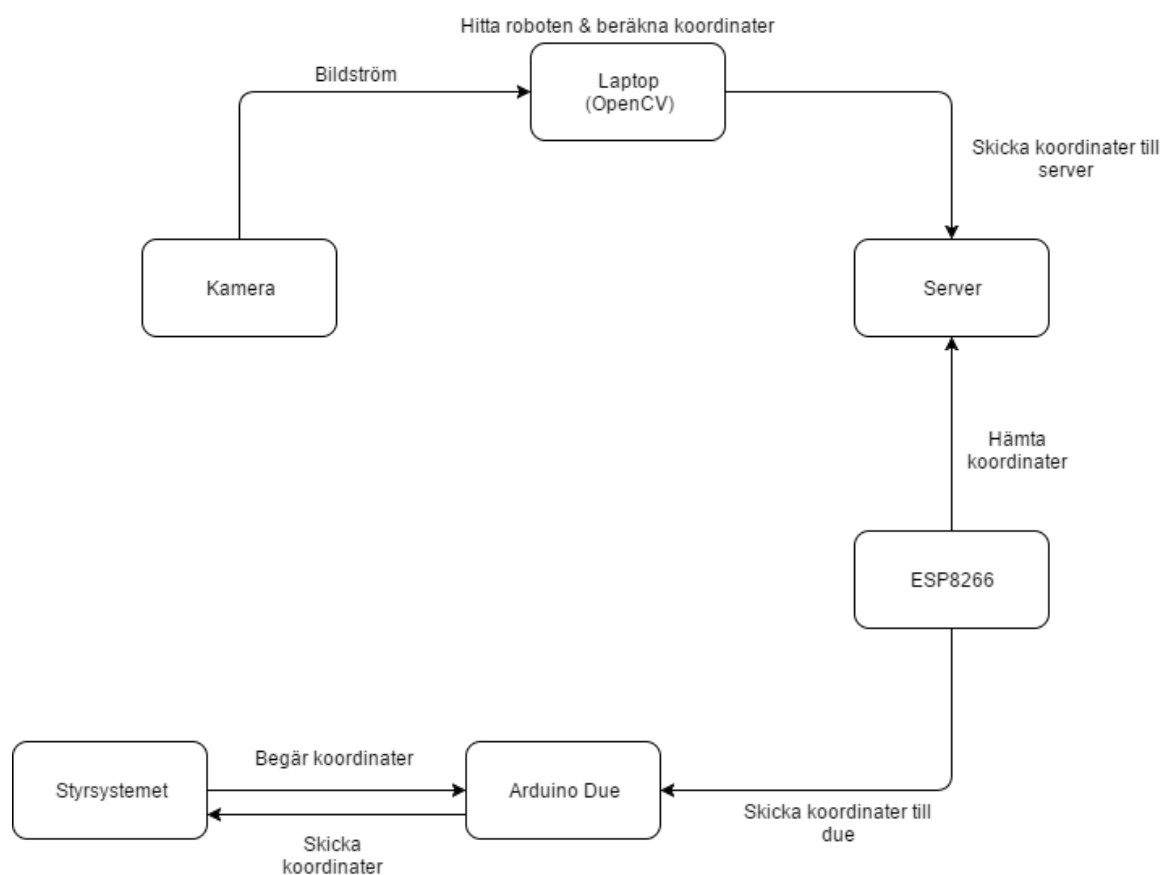


# INFORMATIONSDOKUMENT – POSITIONERINGSSYSTEM

Detta är ett dokument för att ge er (plattformsgrupperna) en bättre förståelse om hur positioneringssystemet fungerar. Men även hur ni ska tolka den datan vi skickar, vad den betyder och hur den relaterar till robotens fysiska position. I detta dokumentet presenterar vi för första lösning vilket använder en kamera och OpenCV för att lokalisera roboten och ta ut dess position. Senare i projektet kommer det att tillkomma ett nytt dokument med en mer avancerad lösning.

Positioneringssystemet som det ser ut idag huvudsakligen av 5 delar, kamera, opencv, server, ESP8266 och ett Arduino Due kort som kommer kommunicera robotens position till ert styrsystem. Kommunikationen mellan de olika enheterna i system illustreras i bilden nedan



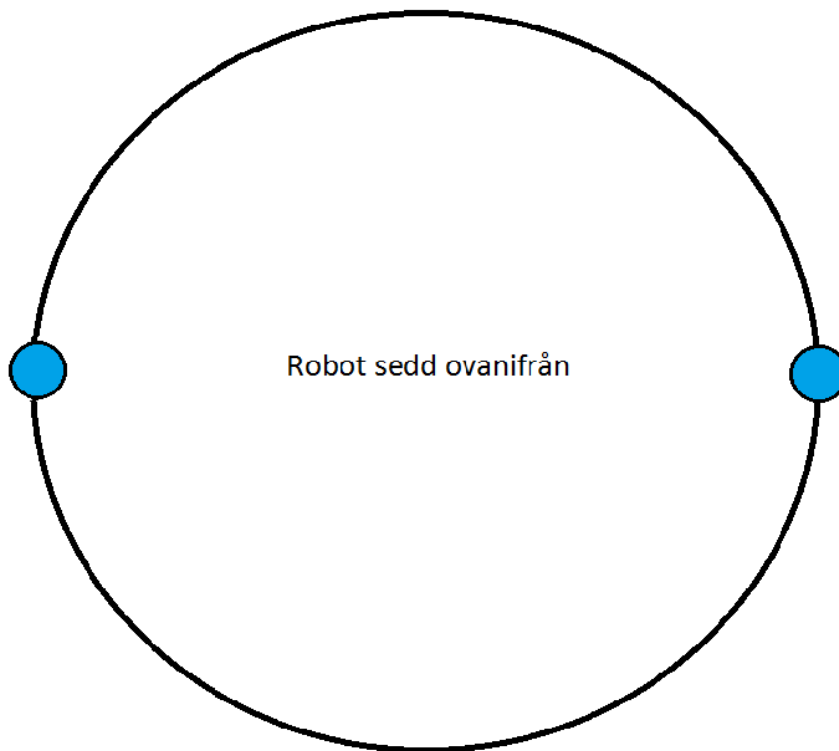
*Figur1: Positioneringssystem*

Det som är relevant för er i systemet i de olika delarna av systemet summeras nedan:

## **OpenCV & Kamera:**

Kameran kommer monteras på tredje våningen på Niagara så det riktas ner mot bottenvåningen där robotarna kommer köra. Banan kommer begränsas till den ytan som kameran lyckas täcka i sin bild, i dagsläget har vi testat med en 5x4m rektangel vilket fungerar väl. Detta kan ändras efter fler tester.

För att identifiera roboten och dess position med hjälp av kameran så använder vi 3 eller 4 lysdioder som monteras på roboten.



I dagsläget använder systemet endast en lysdiod men det kan som sagt tillkomma fler. I fallet att vi använder två lysdioder positionerar vi dessa gärna som i bilden ovan. Dessa lysdioder behöver alltid vara synliga annars kommer OpenCV inte kunna lyckas identifiera roboten och positionen kommer inte kunna kommuniceras. Var på roboten som lysdioderna monteras är flexibelt, d v s ni kan bestämma själv i princip med bilden ovan i åtanke. Men tänk på att informera påbyggnadsgrupperna var ni tänker att lysdioderna ska monteras så inte påbyggnaden täcker lysdioderna.

#### **Server:**

Servern lagrar alla positioner, men lämnar hela tiden ut den senaste positionen till ESP8266, d v s den aktuella positionen för roboten.

#### **ESP8266:**

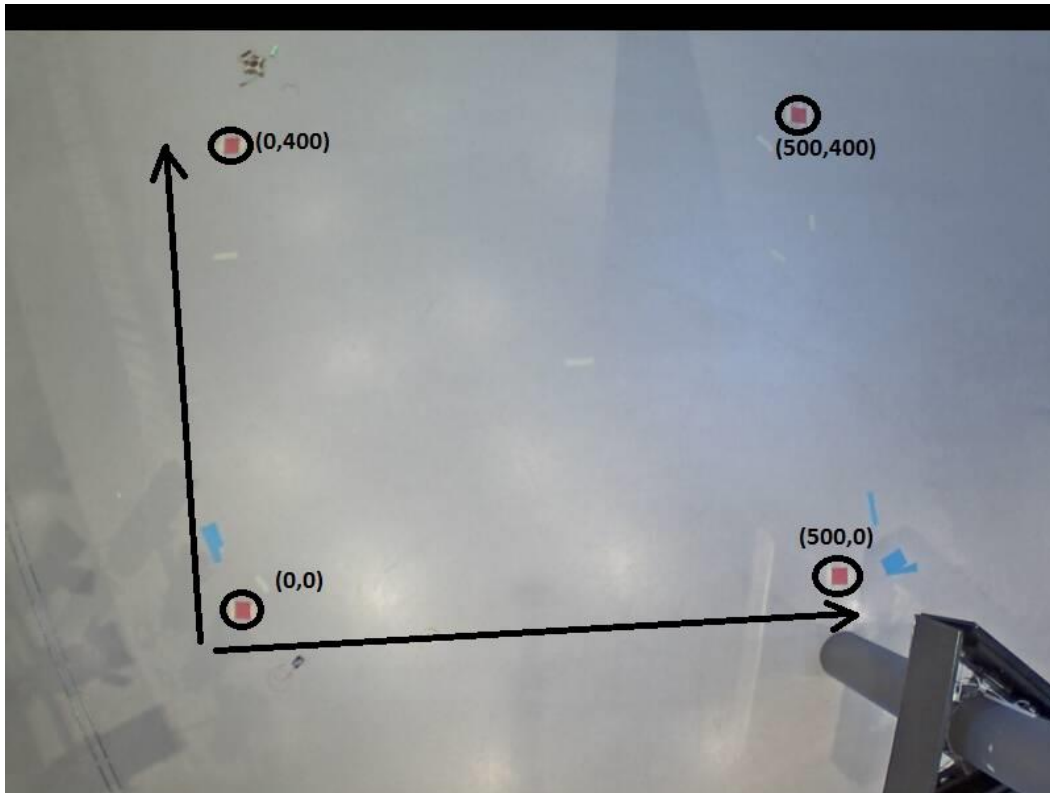
ESP8266 är inte så intressant för er del, den är endast en "gateway" för att få koordinaterna till vår due med hjälp av http bibliotek. ESP:n ligger och hämtar de senaste koordinater hela tiden från servern och kommunicerar dessa till arduino due:n.

#### **Arduino Due:**

Due:ns uppgift är att hela tiden ta emot koordinaterna från ESP:n, nuvarande hastigheter är fyra par av koordinater per sekund som sedan kan skickas vidare till ert styrsystem.

## KOORDINATSYSTEM & KOORDINATERNA:

Koordinatsystemet som pixelkoordinaterna från OpenCV mappas till fungerar på ett sådant sätt att vi mäter upp område t ex ett 5x4m rektangel, där det nedre vänstra hörnet representerar origo. Enheterna i koordinatsystemet är centimeter, detta betyder att det nedre högra hörnet har koordinaterna (500,0) och det övre (500,400). Detta illustreras i följande bild



Figur2: Koordinatsystem på bottenvåning Niagara

De koordinaterna ni kommer få från vårt arduino due är alltså (x,y)-koordinater inom detta området. Testerna vi gjort hittills visar en noggrannhet på 15 - 20 cm. Detta kommer bli mer tydligt när tester med robotarna görs, och något vi jobbar på att förbättra. Med hjälp av dessa koordinaterna hoppas vi att ni ska kunna navigera fram till objekten.

### Dataformat på koordinaterna:

Vårt förslag på hur koordinaterna presenteras är i 16 bitas int. När dessa koordinater skall kommuniceras behöver de vara i bytes och därmed delas varje värde upp i två.

1. `uint16_t x1 = 50;`
2. `Byte x1_H = x1 & 0xFF00;`
3. `Byte x1_L = x1 & 0x00FF;`

Due kortet kommunicerar då en byte array enligt följande.

1. `Byte data[] = [x1_H, x1_L, y1_H, y1_L, x2_H, x2_L, y2_H, y2_L];`