



LUND  
UNIVERSITY

# Lab related seminar

## ETSF15

---

STEFAN HÖST

DEPT. OF EIT, LUND UNIVERSITY



# Idea of two first two labs

---

## Lab 1 (Phy layer)

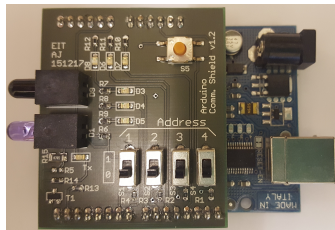
- Point to point communication
- Phy layer based on optics
- Transmit led color
- Load data in frame
- Wait for ACK

## Idea

- Construction of (very) simple network card
- Construction based on an Arduino shield
- Implement behaviour

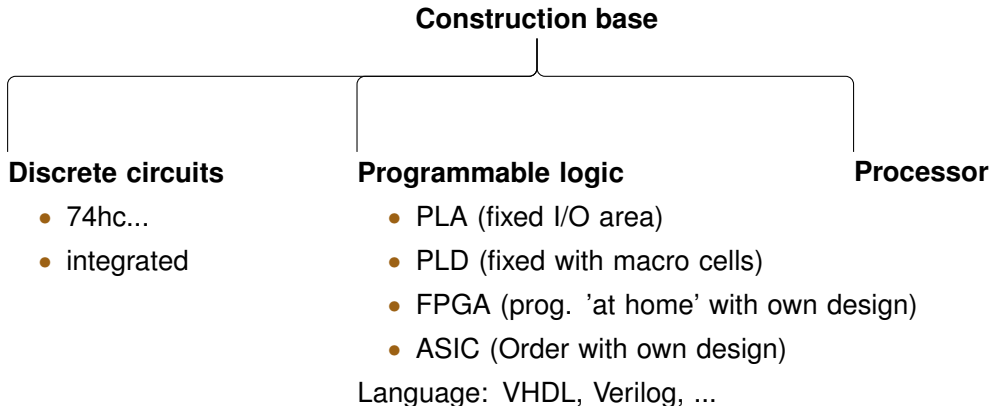
## Lab 2 (Link layer)

- Add access point
- Addressing
- Retransmission scheme
- Error control (CRC)



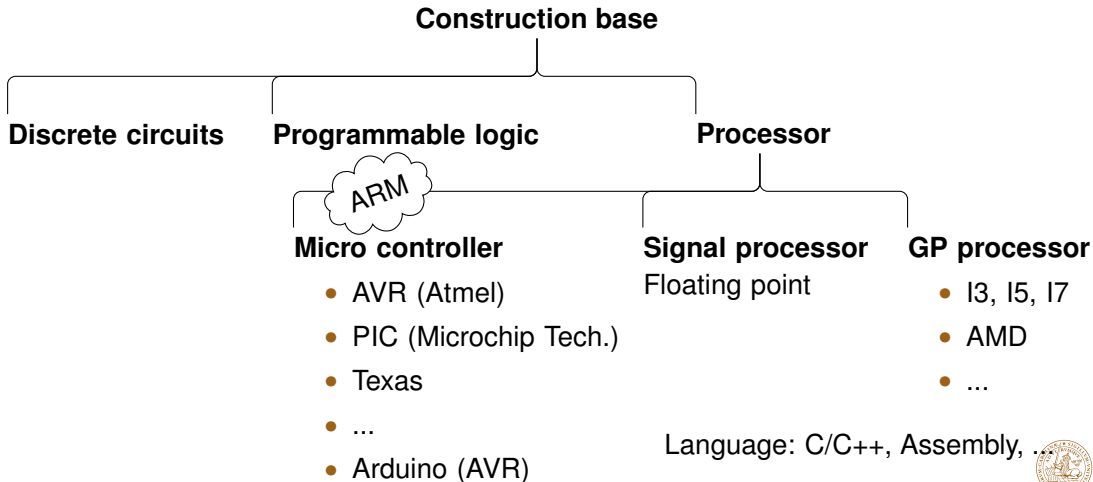
# Hardware constructions

---



# Hardware constructions

---

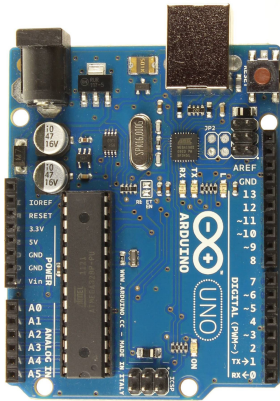


# About programming Arduino

---



# Arduino general code structure



## General code skeleton

```
// global variable and constant declarations

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

# I/O (LEDs)

---

## Blink

```
int LED1=3;

void setup() {
  // put your setup code here, to run once:
  pinMode(LED1,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED1,HIGH);
  delay(300);
  digitalWrite(LED1,LOW);
  delay(300);
}
```

# I/O (Serial monitor)

---

## Serial

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  [...]  
  Serial.print(x); // Print argument  
  Serial.println(x); // Print argument followed by newline  
}
```



# Flow control examples (C/C++)

---

if

```
if (a>b){  
    c = a;  
}else{  
    c = b;  
}
```

(?:) [same res as above]

```
c = (a>b ? a : b);
```

for

```
for (t=0; t<10; t++){  
    Serial.println(t);  
}
```

while

```
t=0;  
while (t<10){  
    Serial.println(t);  
    t++;  
}
```

switch

```
switch (Value){  
    case 0:  
        Serial.println('zero');  
        break;  
    case 1:  
        Serial.println('one');  
        break;  
    default:  
        Serial.println('other');  
        break;  
}
```

# Bit operations (C/C++)

---

## Shift

```
// Shift left k steps  
a = a << k;  
// Shift right k steps  
a = a >> k;
```

## Shift

```
a <<= k;  
a >>= k;
```

## Bit-wise logic

```
// AND  
a = a & 15;  
// OR  
a = a | 15;  
// XOR  
a = a ^ 15;
```

## Bit-wise logic

```
a &= 15;  
a |= 15;  
a ^= 15;
```

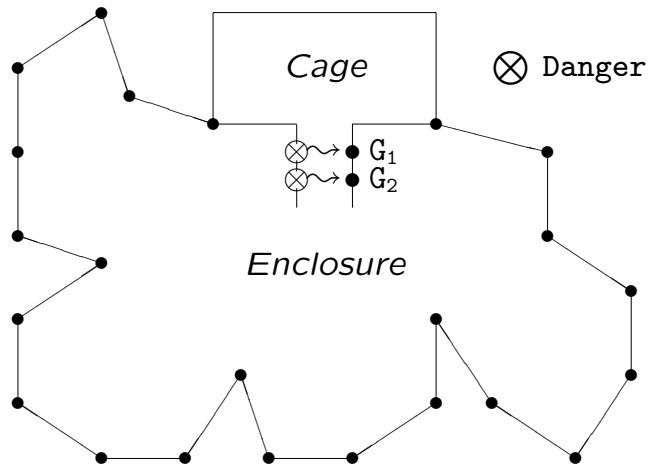
# Lejonburen on Arduino

---



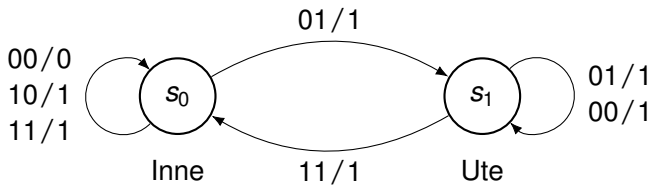
# Lejonburen

---



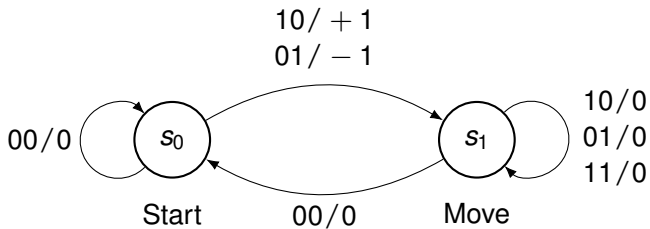
# Graf ett lejon

---



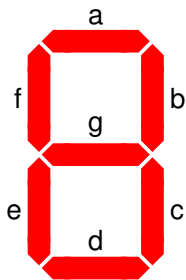
# Graf många lejon

---



# Display driver

---



LED	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1