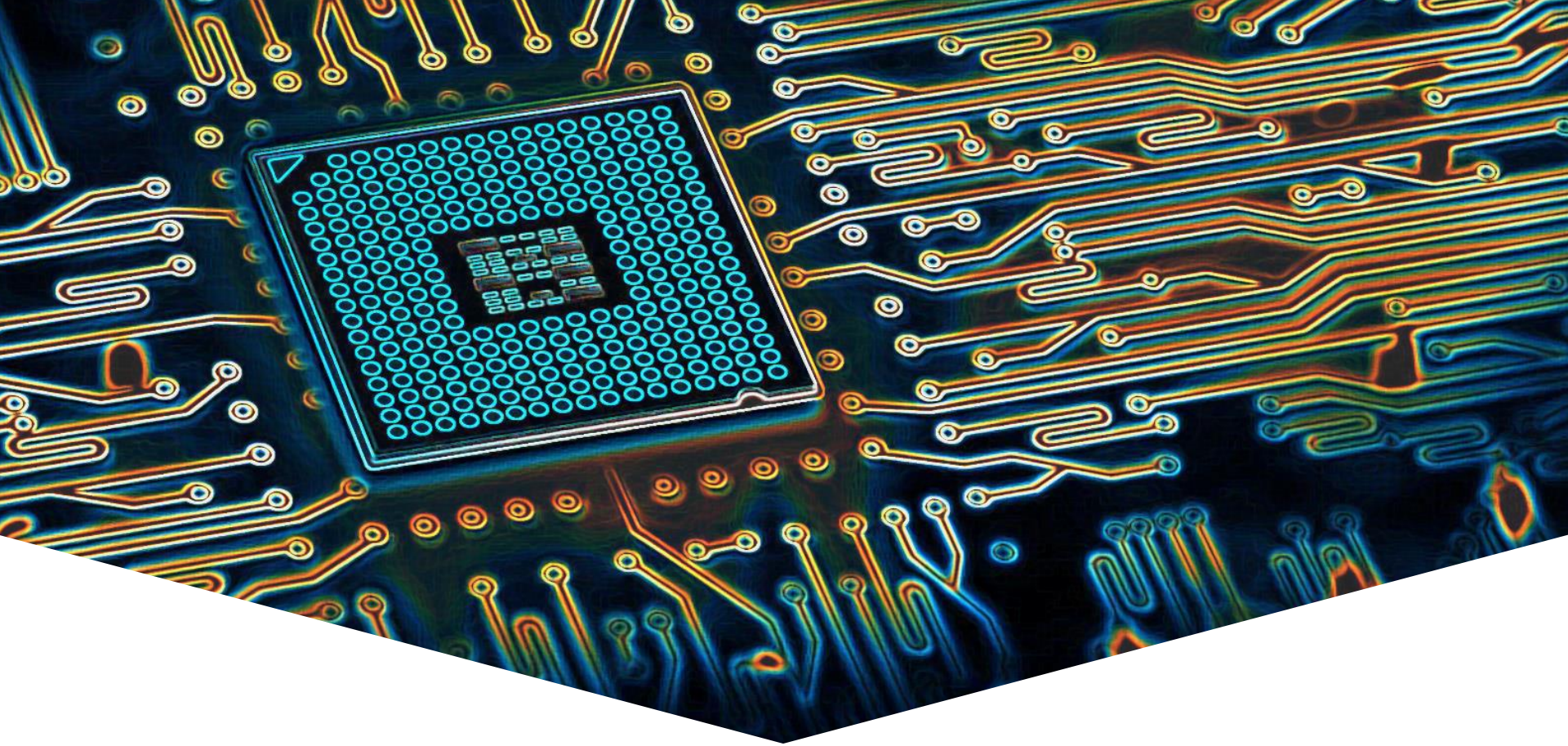




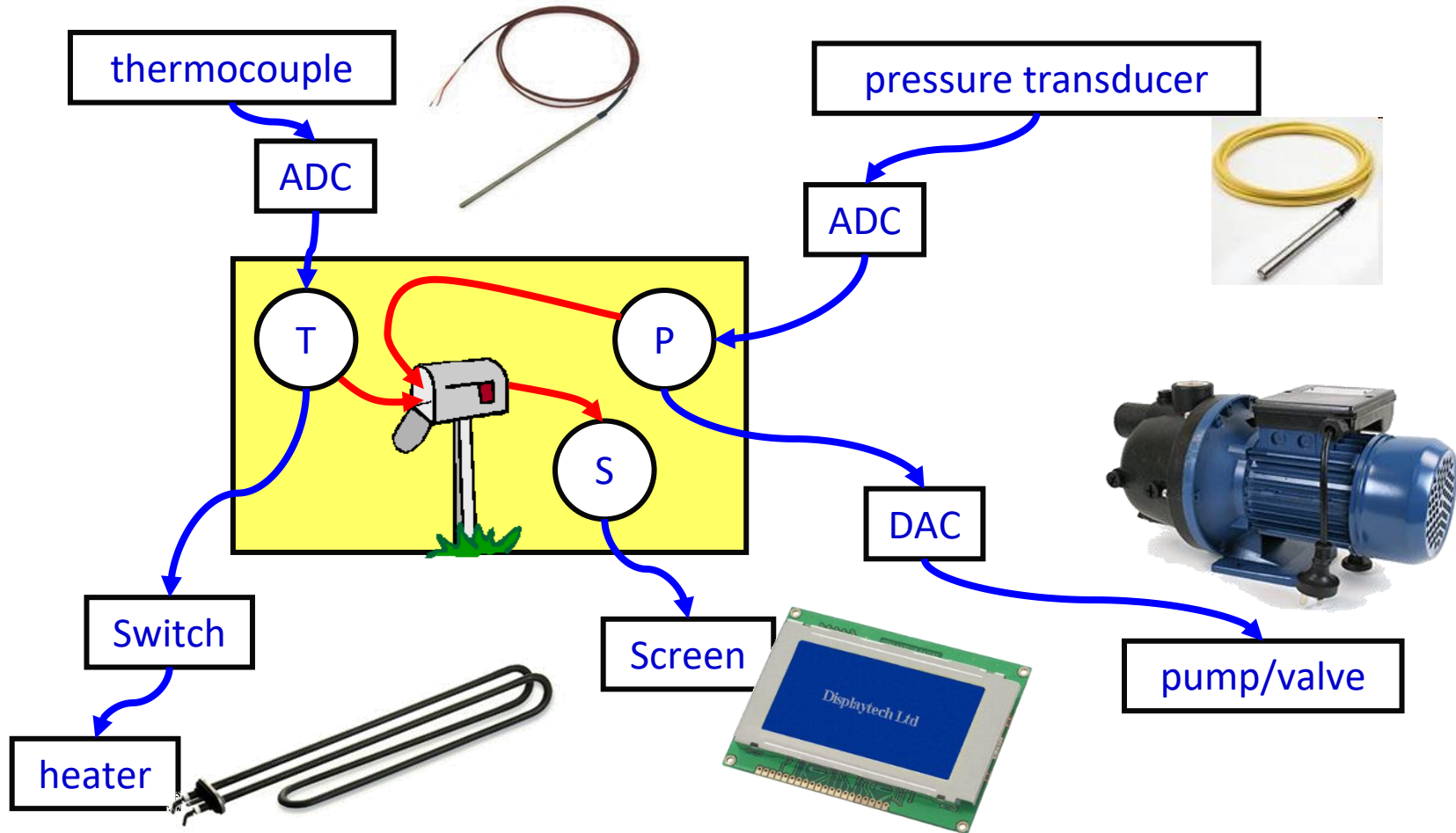
EMS20 Week 4 Lab 2

Leerdoelen week 4 lab 2. Je leert:

- hoe pthreads met elkaar kunnen communiceren met behulp van een message queue (**mqueue**).

Communicatie tussen taken

EMBEDDED SYSTEMS



Communicatie via mqueue (1 van 2)

```
int main(void) {  
    Board_init();  
    mqd_t mqdes;  
    struct mq_attr mqAttrs;  
    mqAttrs.mq_maxmsg = 3;  
    mqAttrs.mq_msgsize = sizeof(int);  
    mqAttrs.mq_flags = 0;  
    mqdes = mq_open("ints", O_RDWR | O_CREAT, 0666, &mqAttrs);  
  
    pthread_t tp, tc;  
    pthread_attr_t attr;  
    pthread_attr_init(&attr);  
    pthread_attr_setstacksize(&attr, 1024);  
    pthread_create(&tp, &attr, &producer, &mqdes);  
    pthread_create(&tc, &attr, &consumer, &mqdes);  
}
```

Maximale aantal berichten.

Maximale aantal bytes per bericht.

Grotere stack zodat printf gebruikt kan worden.

Communicatie via mqueue (1 van 2)

EMBEDDED SYSTEMS

```
void *producer(void *pp) {
    mqd_t mq = *(mqd_t *)pp;
    for (int i = 0; i < 10; i++) {
        mq_send(mq, (char *)&i, sizeof(i), 0);
    }
    return NULL;
}

void *consumer(void *pp) {
    mqd_t mq = *(mqd_t *)pp;
    while (1) {
        int msg;
        mq_receive(mq, (char *)&msg, sizeof(msg), NULL);
        printf("%d\n", msg);
    }
}
```

[source](#)

```
[Cortex_M4_0] Ready... Go!
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Extra producer

```
pthread_t tp1, tp2, tc;  
pthread_attr_t attr;  
pthread_attr_init(&attr);  
pthread_attr_setstacksize(&attr, 1024);  
pthread_create(&tp1, &attr, &producer, &mqdes);  
pthread_create(&tp2, &attr, &producer, &mqdes);  
pthread_create(&tc, &attr, &consumer, &mqdes);
```

```
[Cortex_M4_0] Ready... Go!
```

```
0  
1  
2  
3  
4  
0  
5  
6  
1  
7  
8  
2  
9  
3  
4  
5  
6  
7  
8  
9
```


Aan de slag!

EMBEDDED SYSTEMS

Aan de slag met [Opdrachten Week 4 Lab 2.pdf](#)

