

# 1 Testresultater

Da systemet blev udviklet, var det vigtigt at teste koden og komponenter, som tingene blev kodet og udviklet. Til at starte med blev de enkelte dele af systemet teste hver for sig, så man var sikker på, den enkelte del virkede, før den blev koblet sammen med andre dele. F.eks. blev værdierne fra LM75 testet via seriel forbindelse gennem RS232 til en computer. Med programmet *Tera Term*, kan man se, om der kommer nogle værdier ud, og om de passer nogenlunde overens med realiteten. De andre dele af systemet (GSM, LCD162 og Buzzer), blev også testet individuelt inden nogle tasks blev sat sammen.

Når alle tasks virkede individuelt, var det næste skridt at teste nogle samlet. For at kunne holde øje med, hvor i koden man befandt sig, når der opstod fejl, var det brugbart at bruge LED-driveren. Denne indeholder en simpelt toggle funktion, som får en bestemt LED til at skifte når denne kaldes. En test kan ses for funktionen `ModemOutput` på Listing 1. Hvis programmet skulle gå i stå et sted, i f.eks. en deadlock, er det også muligt at se, hvor i programmet man er kommet til via LEDs.

Alle individuelle tests, samt enkelte sammensatte tests virkede efter hensigten. Problemet kom først, da alt blev sat sammen. Det var dog ikke rigtigt muligt at teste her, pga. hukommelses plads. Selvom nogle ændringer blev lavet i `FreeRTOSconfig.h` filen, så det kunne compile, ville programmet slet ikke starte på Mega32-processoren.

For at sikre, at den indbyggede timing i FreeRTOS, ikke blev ændret af tilføjet kode, med tanke på TIMERS, blev der oprettet en simpel task. Den task havde kun én funktion, nemlig at toggle LED 0 hvert 100 millisekund. Herved var det hurtigt at opdage, hvis programmet fra start ikke kørte, hvis programmet pludseligt ændrede timing eller programmet gik i stå. Denne task blev slettet, da hukommelsespladsen løb op.

Listing 1: `ModemOutput` med LED-test

```
1 void ModemOutput(void *pvParameters)
  {
3   char buffer = 0;
   char flag = 1;
5
   while(1)
7   {
       while(!CharReady())
```

```

9      {
10         vTaskDelay(100);
11         toggleLED(1);
12     }
13     LCDGotoXY(0,1);

15     while(flag)
16     {
17         toggleLED(2);
18         buffer = ReadChar();
19         if(buffer == '\r')
20         {
21             flag = 0;
22         }
23         LCDDispChar(buffer);
24     }

25     flag = 1;
26     toggleLED(3);
27     vTaskDelay(50);
28 }
29 }

```