

# Routenstatistik von Segelregatten

**Maturaprojekt**  
**5AHI 2001/2002**

# Was soll das Projekt können?

- Erfassung der Bootspositionen
- Erfassung von Windgeschw. und Richtung auf jedem Boot
- Grafische Darstellung der
- Bootsbewegungen
- Errechnung des wahren Windes

# Technologien

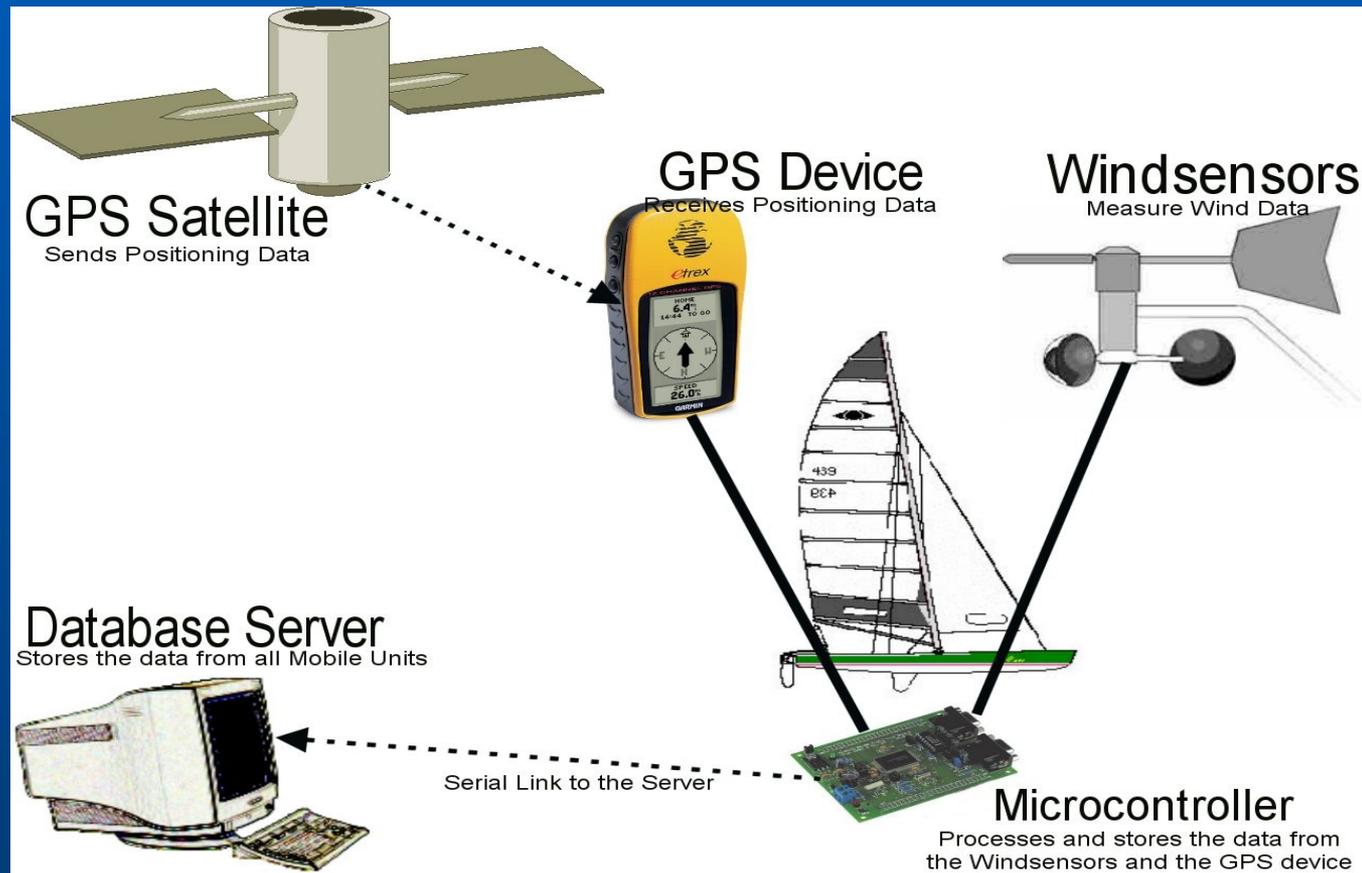
- **Positionserkennung über GPS**
- **Mikrocontroller:  $\mu$ C167**
- **Programmiersprache: Java, ANSI-C**

# Gliederung in Teilbereiche

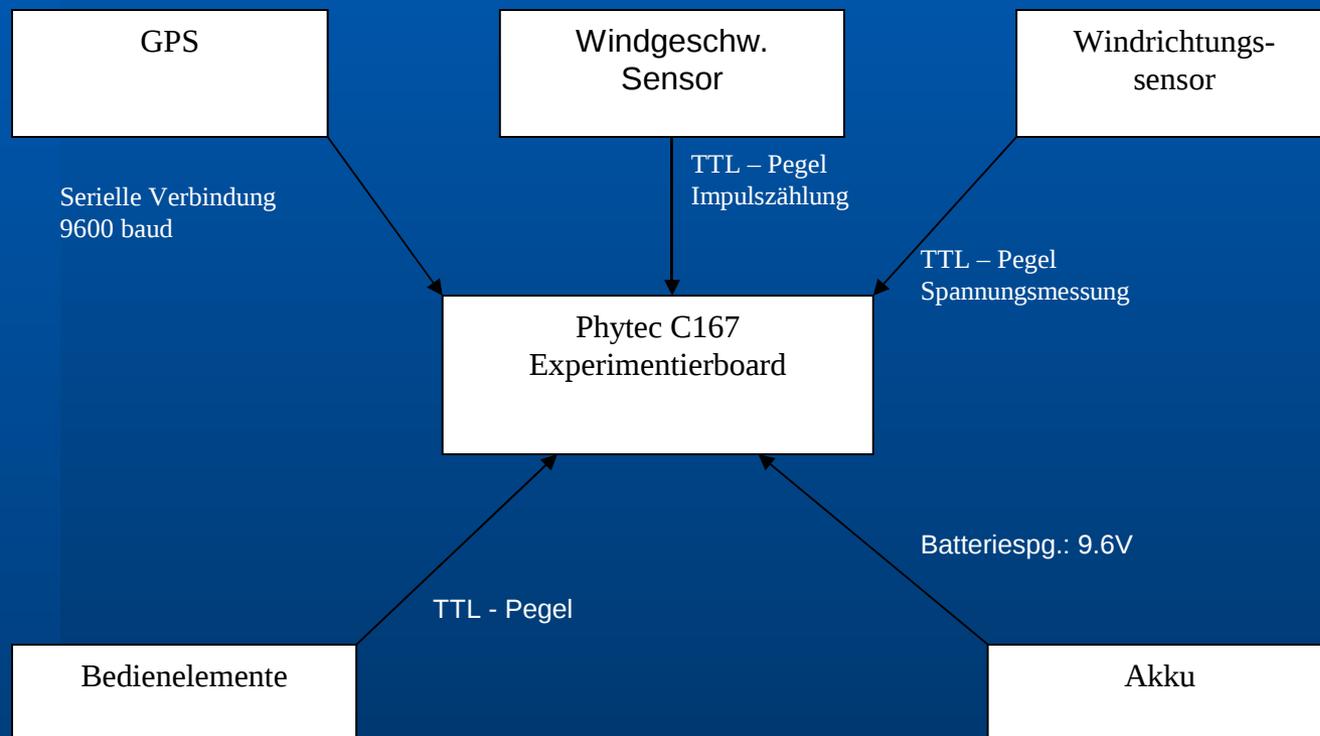
---

- **Mobile Unit**
- **PC Software**
- **Diplomarbeit: GPS Datenauswertung**
- **Diplomarbeit: Visualisierung**

# Blockschaltbild des Gesamtsystems



# Blockschaltbild der Mobile Unit



# GPS Simple Text Protokoll

Vom GPS gelieferte Daten:

@	011024143816	N4637399	E01402396	S	015	<u>+00399</u>	E0000	N0056	<u>U0000</u>
Start	Time data	Latitude	Longitude		Error	Altitude	E/W vel.	N/S vel.	Vertical vel.
					GPS Status				

Die rot unterlegten Felder werden von uns nicht benötigt.  
Daraus ergibt sich das von uns verwendete Protokoll

# Datenspeicherung

- 1.) Parsen der GPS Daten
- 2.) Teile des Strings in einen Buffer kopieren
- 3.) Datentyp umwandeln
- 4.) In einer Struktur speichern
- 5.) Datenstruktur wird sequentiell in einem Array gespeichert

# Datenstruktur

22 25 10 18 47 N 4636665 E 1402810 S 15 E 0 N 0 270 125

struct data

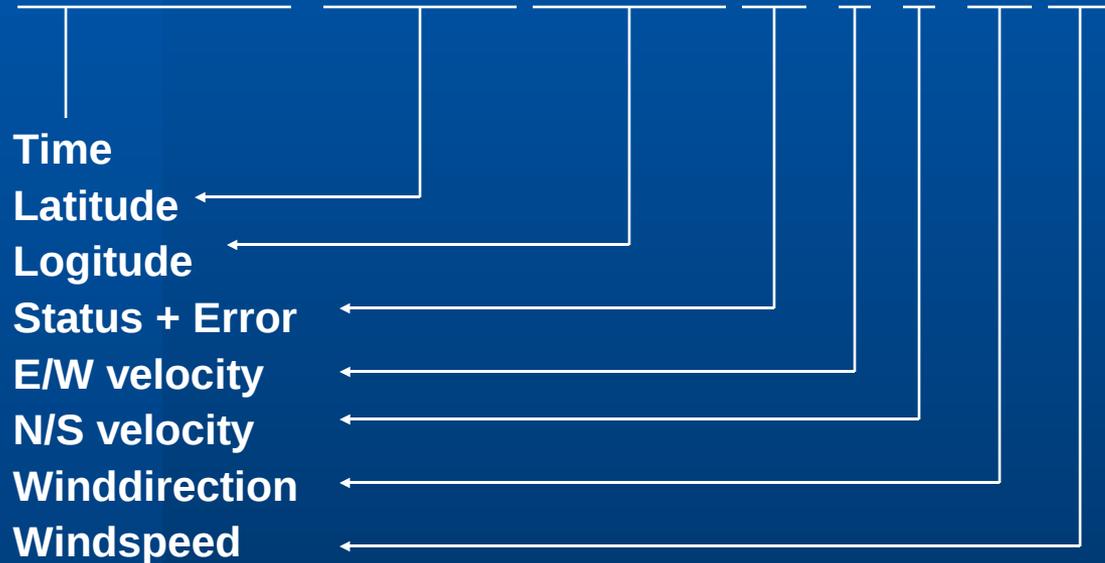
```
{
    short year,month,day,hour,minute,second; //GPS Time Data
    unsigned long lap;                        //Latitude position

    unsigned long lop;                        //Longitude position
    char lah;                                 //Latitude Hemisphere
    char loh;                                 //Longitude Hemisphere
    char ps;                                  //Position Status
    char ewvd;                                //East-West velocity direction
    short hper;                               //Horizontal position error
    short ewvm;                               //East-West velocity magnitude
    short nsvm;                               //Noth-South velocity magnitude
    short wd;                                 //Winddirection
    short wvm;                               //Wind velocity magnitude
    char nsvd;                                //Wind North-South velocity direction
```

# Mobile Unit Simple Text Protokoll

Von der Mobile Unit gelieferte Daten:

2.2.25.10.18.47.N4636665.E1402810.S15.E0.N0.270.125



# Speicherbedarf

ungekürzte GPS Daten + Winddaten (String):  $56 + 6 = 62$  Byte  
3,72 kByte / min  
223,2 kByte / h  
-> 1,116 MByte in 5 Stunden

gekürzte GPS Daten + Winddaten (String):  $44 + 6 = 50$  Byte  
3 kByte / min  
180 kByte / h  
-> 900 kByte in 5 Stunden

gekürzte GPS Daten + Winddaten (Struktur): 40 Byte  
2,4 kByte / min  
144 kByte / h  
-> 720 kByte in 5 Stunden

Wenn Abtastung im Sekundentakt

# Speicherbedarf

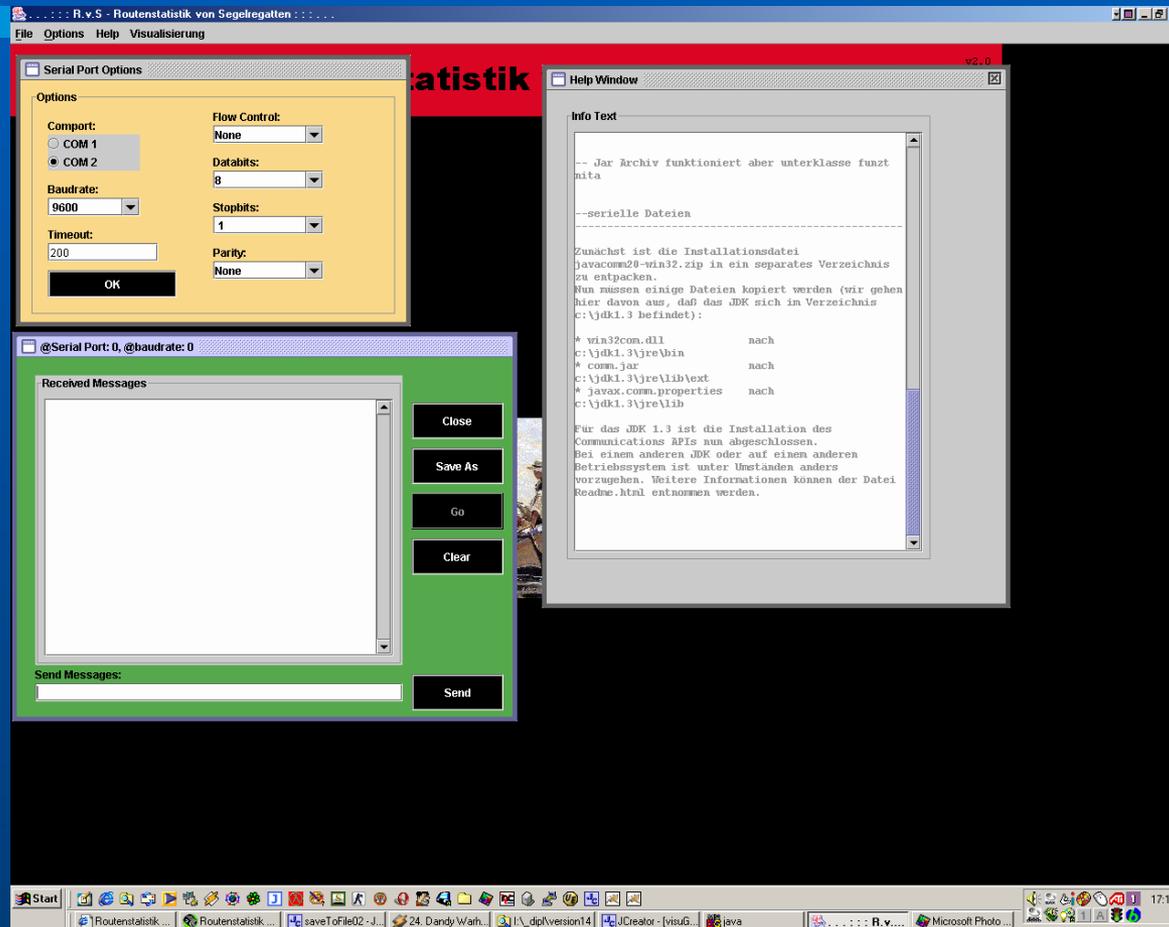
## Zu Beachten:

- **Word Alignment: Durch Plattformabhängige Speicherorganisation unterschiedliche Strukturgrößen (Windows: 37Byte, C167: 36Byte)**
- **Durch ändern der Abtastrate, kann der Speicherbedarf zusätzlich eingeschränkt werden**

**Abtastrate 5s:           144kByte in 5 Stunden**

**Abtastrate 10s:       72kByte in 5 Stunden**

# PC - Software



# PC - Software

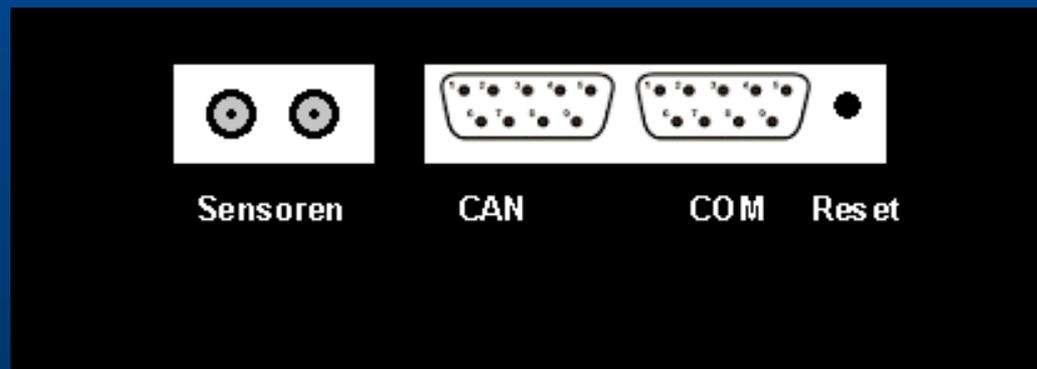
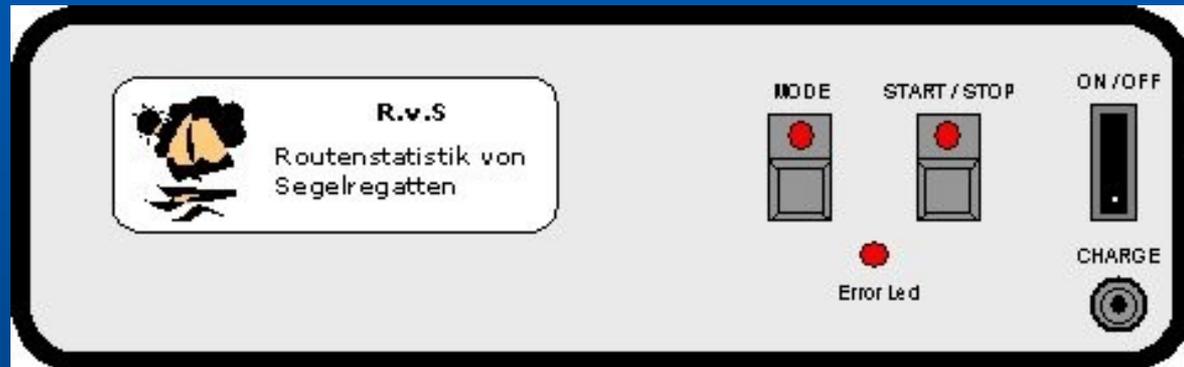
```
public void init() throws Exception {
    Enumeration en = CommPortIdentifier.getPortIdentifiers();
    while (en.hasMoreElements()) {
        CommPortIdentifier cpi = (CommPortIdentifier)en.nextElement();
        if (cpi.getPortType() == CommPortIdentifier.PORT_SERIAL) {

            if (cpi.getName().equals("COM"+comPort)) {
                com = (SerialPort)cpi.open( "Com"+comPort, timeout );
                com.setSerialPortParams(baudrate,databits,stopbits,parity);
                com.setFlowControlMode(SerialPort.FLOWCONTROL_NONE);
                serOut = com.getOutputStream();
                serIn = com.getInputStream();
                it = new InputThread(serIn, txtAusgabe);
                it.start();
            }
        }
    }
}
```

# Input Thread

```
public void run() {
    StringBuffer s; int c;
    while (inputRun) {
        try {
            s = new StringBuffer();
            String line="";
            while ((c = serIn.read()) != '\n' ) {
                s.append((char)c);
            }
            line = s.toString();
            if(go) {
                f.write(line+"\n");
            }
            txt.append(line+"\n");
        } catch(Exception e) {
            System.err.println("Error in serial read!!! \n"+e.toString());
        }
    }
}
```

# mobile Unit



# Anwendung aufzeichnen

- Sensoren an mobile Unit anschließen
- Einschalten der mobile Unit
- Mittels Mode Taster Record Mode auswählen
- Anhand Start/Stop Taster Aufzeichnung starten.
- Nach der Regatta Aufzeichnung stoppen.

# Anwendung einlesen

- mobile Unit an PC anschließen
- Programm starten
- Im Programm zuerst Optionen einstellen
- Anhand New eine neue Verbindung aufbauen
- Datei auswählen
- Auf beiden Seiten die Übertragung starten

# Ende

---

**DANKE FÜR IHRE AUFMERKSAMKEIT**