# Vorlesung
# Automotive Software Engineering

## Aktuelles und Organisatorisches

TU Dresden, Fakultät Informatik

Sommersemester 2012

Prof. Dr. rer. nat. Bernhard Hohlfeld

bernhard.hohlfeld@daad-alumni.de

# Vorlesung Automotive Software Engineering

- Technische Universität Dresden,
  Fakultät Informatik, Professur Softwaretechnologie
  http://st.inf.tu-dresden.de/content/index.php?node=teaching
- Sommersemester 2012

| | |
|---|---|
| Vorlesungszeiten | 6 Blöcke à 3 h entspricht 18 Zeitstunden<br>Exkursion à 4,5 h<br>insgesamt 22,5 h oder 2 SWS |
| Termine | Montag, 14:50 – 16:20, 16:40 – 18:10 (5. und 6. DS)<br>23. April, 7. Mai, 21. Mai, 18. Juni, 25. Juni, 9. Juli<br>Ersatztermine: 14. Mai, 11. Juni, 2. Juli<br>Exkursion Gläserne Manufaktur: <tbd>, Dienstag nach Vorlesung<br>Prüfung <tbd, Kernprüfungszeit bis Sa, 11.08.2012> |
| Raum | INF E009 |
| Dozent | Prof. Dr. rer. nat. Bernhard Hohlfeld<br>bernhard.hohlfeld@daad-alumni.de |
| Betreuer | Dipl.-Inf. Florian Heidenreich<br>florian.heidenreich@tu-dresden.de |
| <Semantik> | Noch festzulegen bzw. ohne Bestätigung aus Vorjahr übernommen |

# Exkursion „Gläserne Manufaktur“

- Werksbesichtigung „Gläserne Manufaktur“ in Dresden
  (Montage Volkswagen Phaeton)
- http://www.glaesernemanufaktur.de
  - Termin <tbd>
  - Treffpunkt <tbd> am Besuchereingang Strassburger Platz / Lennéstrasse
    Gleicher Eingang wie Café und Restaurant
  - Kosten: 45 EURO für eine Gruppe (Wurde bisher vom Lehrstuhl übernommen)
  - Teilnehmer: max. 20 Personen
  - Anmeldung per email an bernhard.hohlfeld@daad-alumni.de:
    - Name, Vorname, email-Adresse
    - Anmeldungen werden in der Reihenfolge des Eingangs berücksichtigt
  - WS 2008/09 ca. 24 Teilnehmer, SS 2010 und SS 2012 ca. 15 Teilnehmer
  - Teilnehmerliste
- Auszug aus Anmeldung:
  hiermit bestätige ich die Führung unserer Gruppe am 12. Juli 2011 um 09:30 Uhr mit 15 Personen in deutscher Sprache zum Preis von 45,00 Euro.
  Ebenfalls wurde ich informiert, dass das Fotografieren und Filmen im Besucherforum der Gläsernen Manufaktur gestattet ist, nicht aber in den Bereichen der Fertigung.

# Seminar Cyber-Physical Systems

- http://st.inf.tu-dresden.de/content/index.php?node=teaching
- Dienstage nach Vorlesung vormittags

# Begriffsklärungen

| Begriff | Beschreibung | Beispiel Automobil | Beispiel Bahn |
|---|---|---|---|
| Embedded System | Rechnergesteuertes technisches System; Kommunikation betrifft nicht die Kernfunktionen | Automobil nach Stand der Technik; Kommunikation für Telematik (Telefonie, Radio, Navigation) | Schienenfahrzeug ohne Leittechnik (Nebenstrecken, Strassenbahnen); nicht mehr Stand der Technik |
| Cyber-Physical System | Mehrere rechnergesteuerte technische Systeme, die untereinander und mit ihrer Umwelt kommunizieren; Kommunikation betrifft die Kernfunktionen | Mehrere Automobile mit Car-to-X-Kommuni-kation incl. Infrastruktur (Zentralen, Baken, Diensteanbieter, ...) | Mehrere Schienenfahrzeuge incl. Leittechnik (onboard und offboard) |

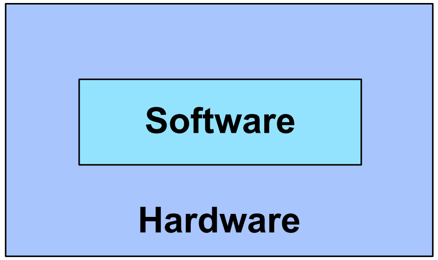
# Onion ring like structure of CPS (modified)

Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)

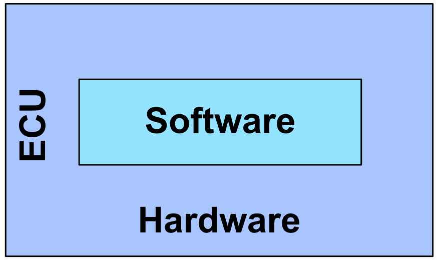
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



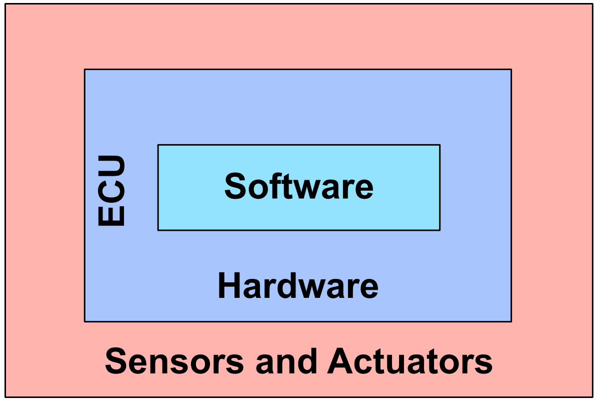
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



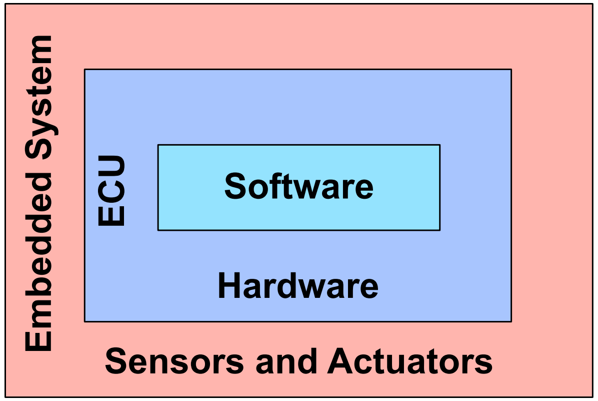
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



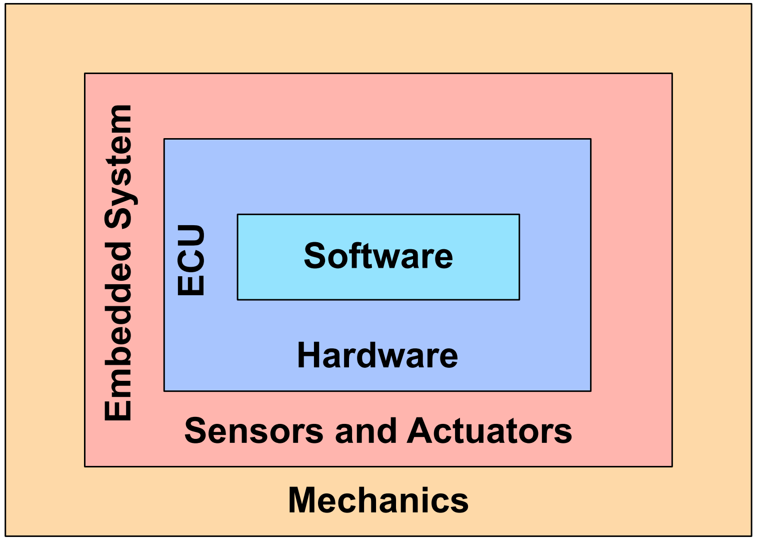
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



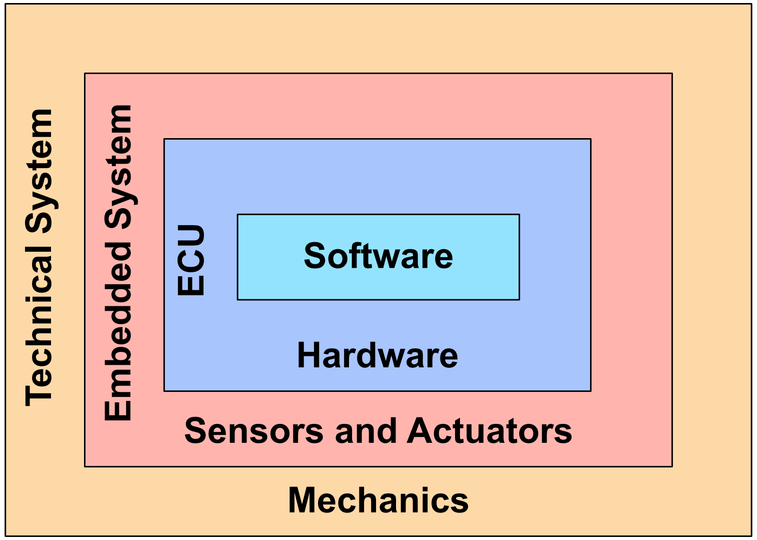
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



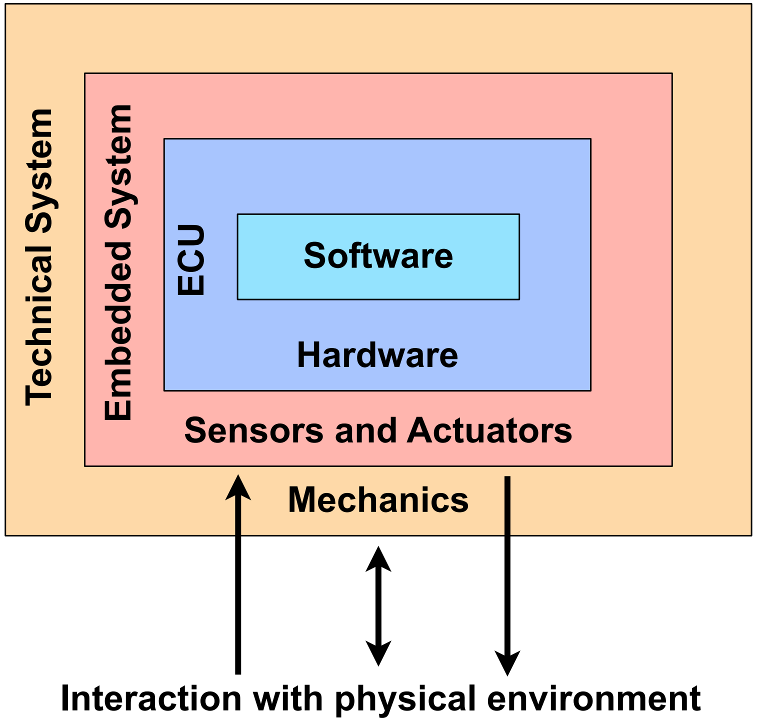
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



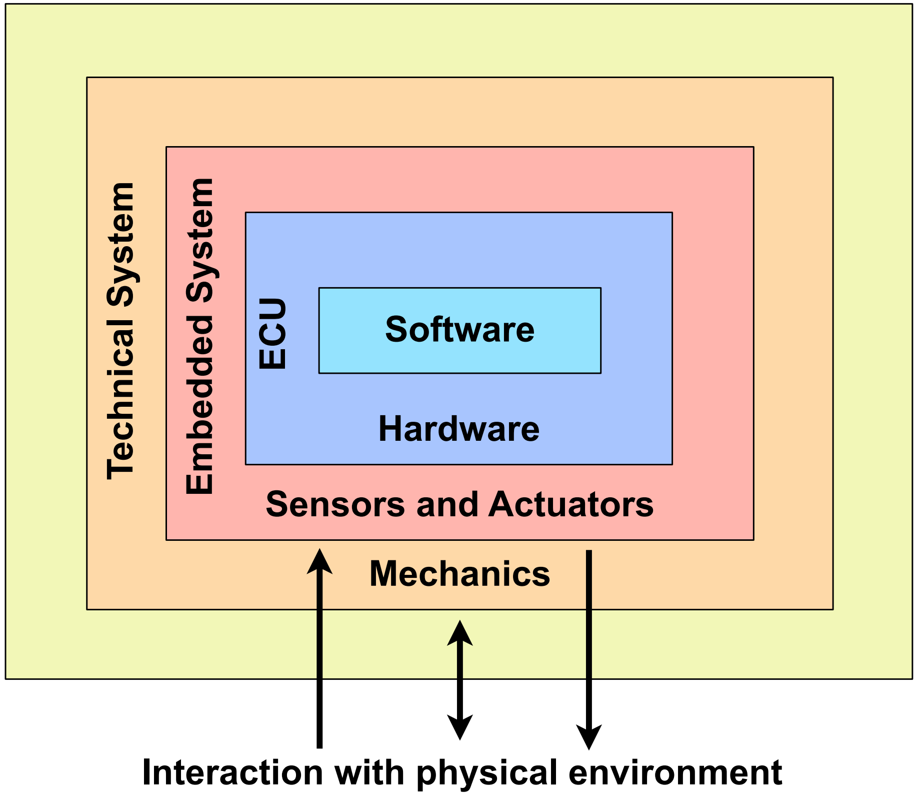
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



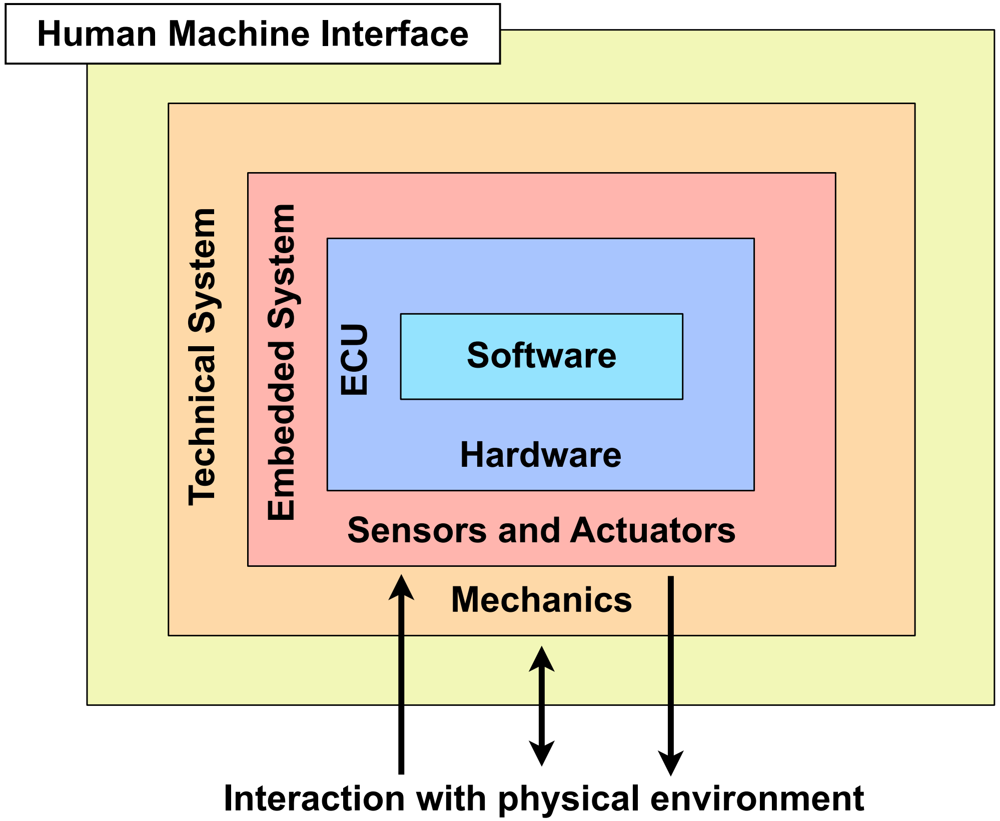
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



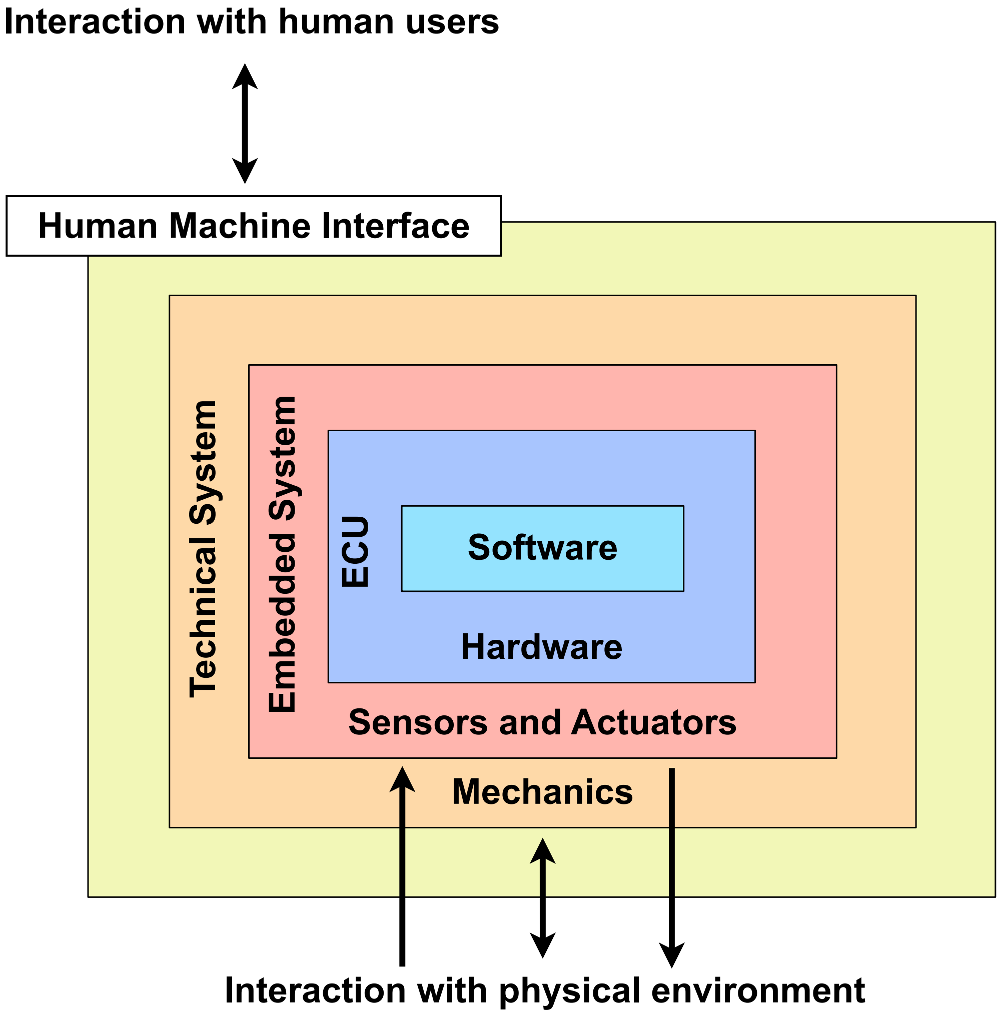
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



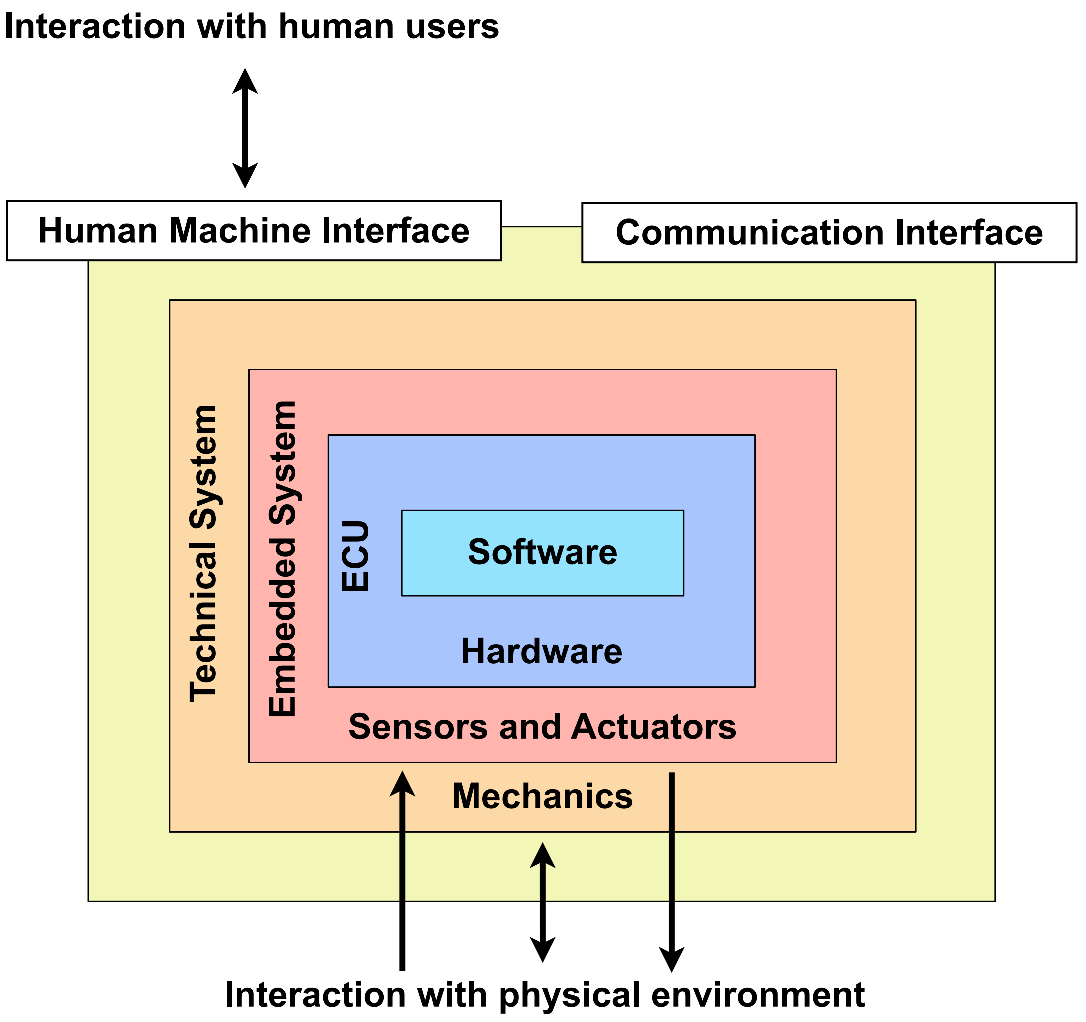
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



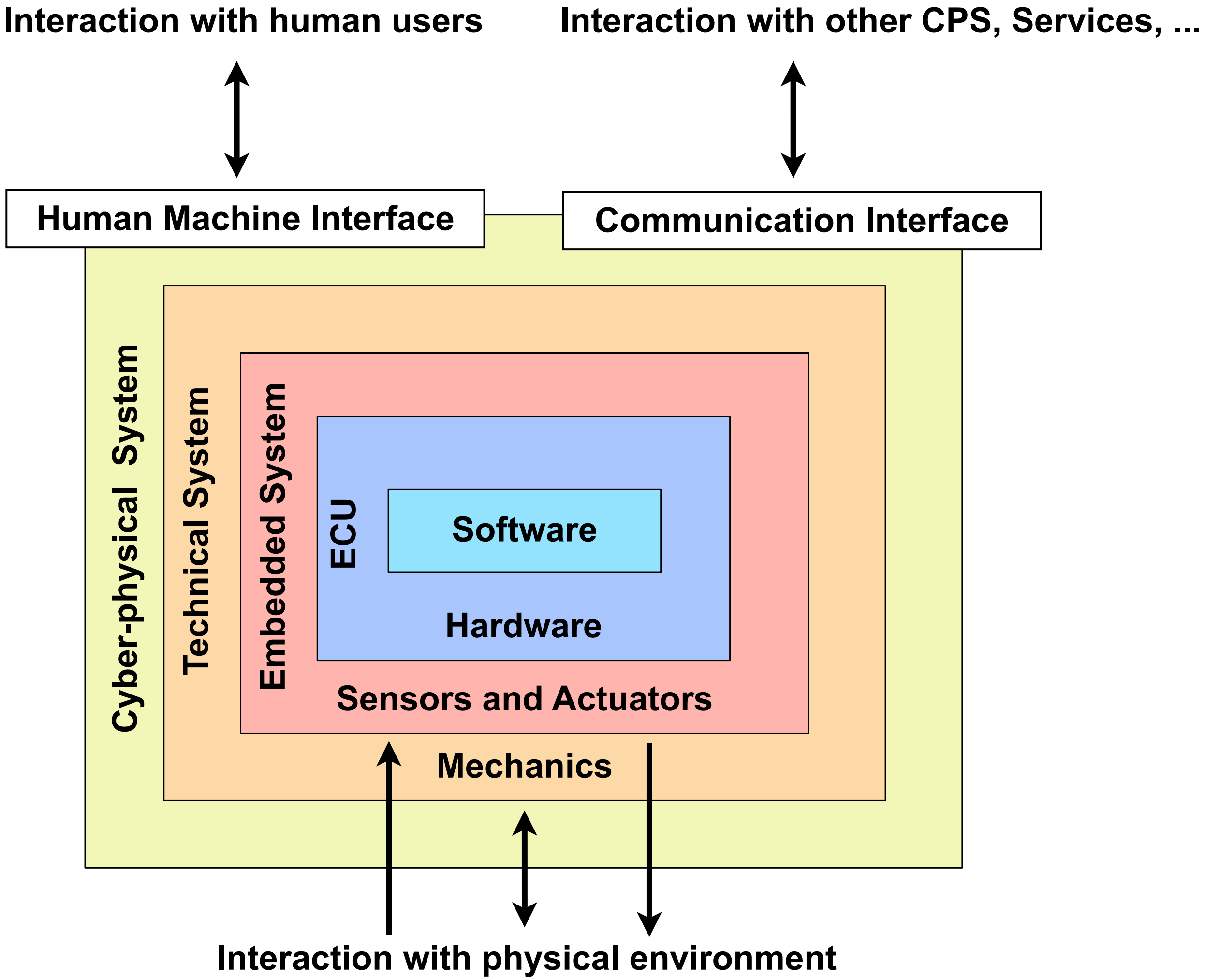
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified)



Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS

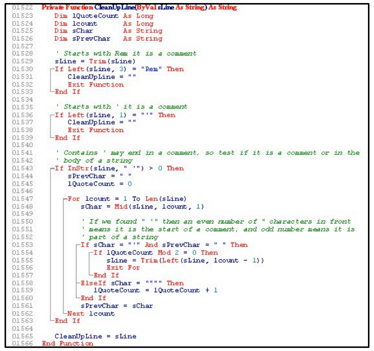
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS

**Wiper Control Software**

Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS

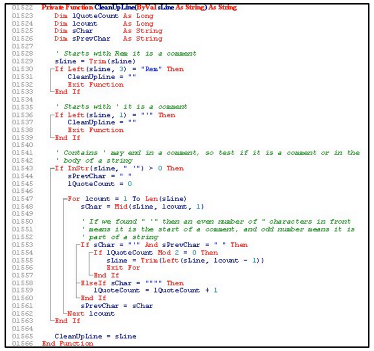


```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

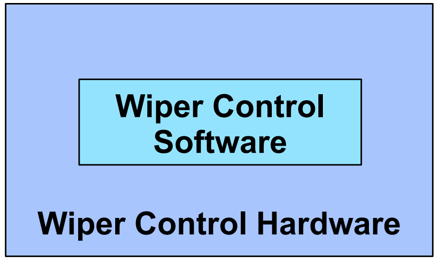
**Wiper Control Software**

Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS

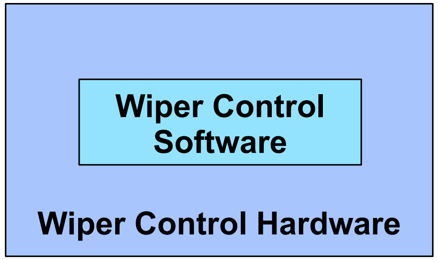


```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```



Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS



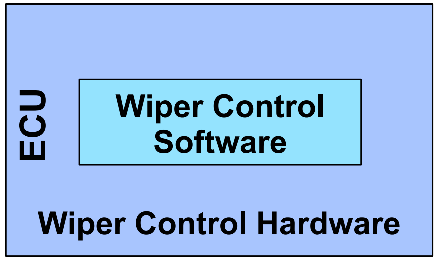
Wiper Control Software

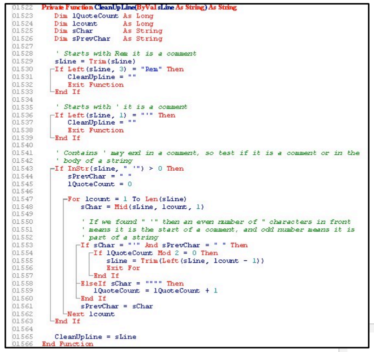
Wiper Control Hardware

```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
   ┌If Left(sLine, 3) = "Rem" Then
   │   CleanUpLine = ""
   │   Exit Function
   └End If

    ' Starts with ' it is a comment
   ┌If Left(sLine, 1) = "'" Then
   │   CleanUpLine = ""
   │   Exit Function
   └End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
   ┌If InStr(sLine, " '") > 0 Then
   │   sPrevChar = " "
   │   lQuoteCount = 0
   │
   │  ┌For lcount = 1 To Len(sLine)
   │  │   sChar = Mid(sLine, lcount, 1)
   │  │
   │  │   ' If we found " '" then an even number of " characters in front
   │  │   ' means it is the start of a comment, and odd number means it is
   │  │   ' part of a string
   │  │  ┌If sChar = "'" And sPrevChar = " " Then
   │  │  │  ┌If lQuoteCount Mod 2 = 0 Then
   │  │  │  │   sLine = Trim(Left(sLine, lcount - 1))
   │  │  │  │   Exit For
   │  │  │  └End If
   │  │  ├ElseIf sChar = """" Then
   │  │  │   lQuoteCount = lQuoteCount + 1
   │  │  └End If
   │  │   sPrevChar = sChar
   │  └Next lcount
   └End If

    CleanUpLine = sLine
End Function
```

Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS

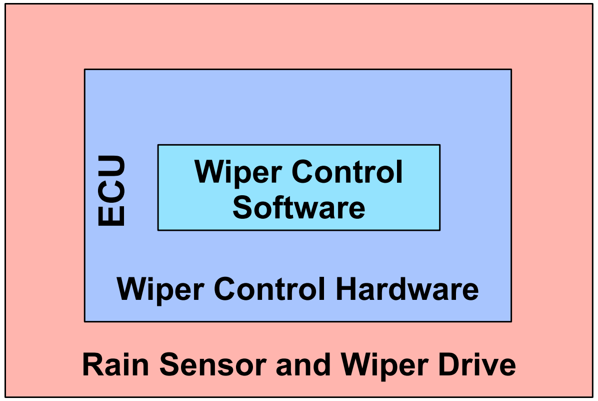




```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

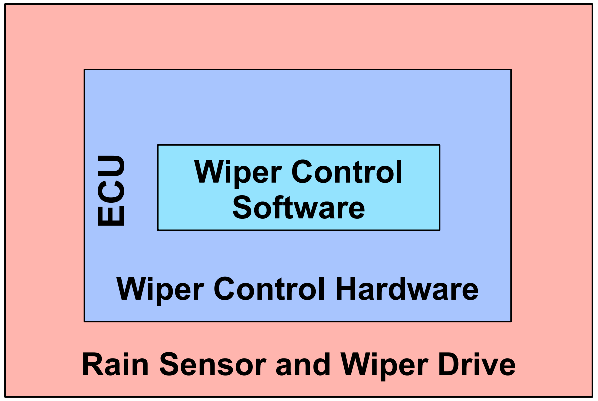
# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS



```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS

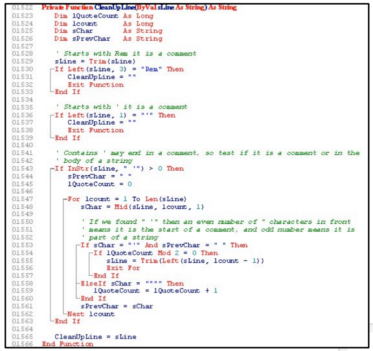


ECU

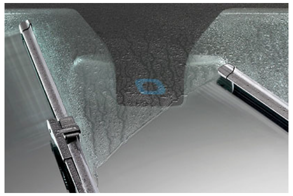
Wiper Control Software

Wiper Control Hardware

Rain Sensor and Wiper Drive

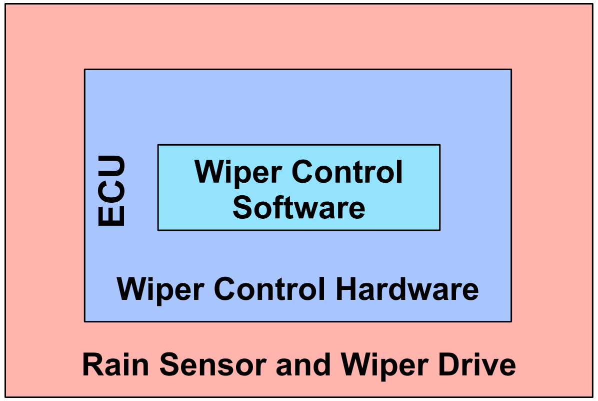


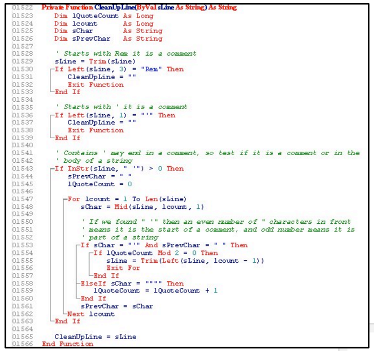
```
Private Function CleanUpLine(ByVal sLine As String) As String
   Dim lQuoteCount As Long
   Dim lcount      As Long
   Dim sChar       As String
   Dim sPrevChar   As String

   ' Starts with Rem it is a comment
   sLine = Trim(sLine)
  If Left(sLine, 3) = "Rem" Then
      CleanUpLine = ""
      Exit Function
  End If

   ' Starts with ' it is a comment
  If Left(sLine, 1) = "'" Then
      CleanUpLine = ""
      Exit Function
  End If

   ' Contains ' may end in a comment, so test if it is a comment or in the
   ' body of a string
  If InStr(sLine, " '") > 0 Then
      sPrevChar = " "
      lQuoteCount = 0

     For lcount = 1 To Len(sLine)
         sChar = Mid(sLine, lcount, 1)

         ' If we found " '" then an even number of " characters in front
         ' means it is the start of a comment, and odd number means it is
         ' part of a string
        If sChar = "'" And sPrevChar = " " Then
           If lQuoteCount Mod 2 = 0 Then
               sLine = Trim(Left(sLine, lcount - 1))
               Exit For
           End If
        ElseIf sChar = """" Then
           lQuoteCount = lQuoteCount + 1
        End If
         sPrevChar = sChar
     Next lcount
  End If

   CleanUpLine = sLine
End Function
```



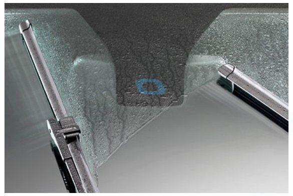
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS



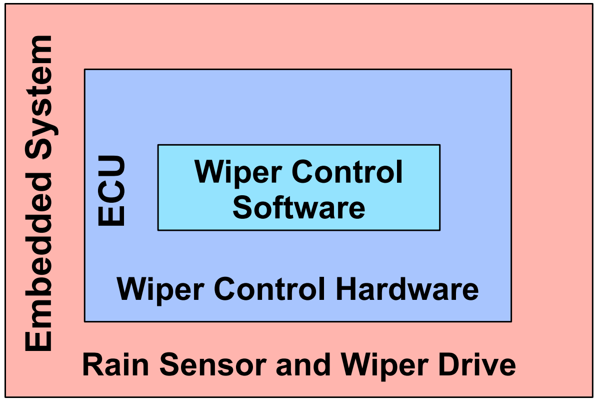


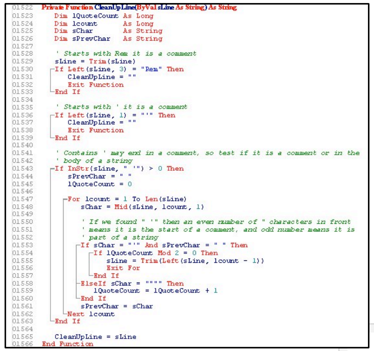
```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```



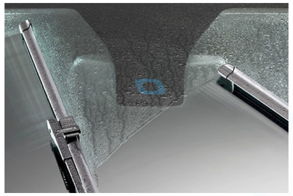
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS



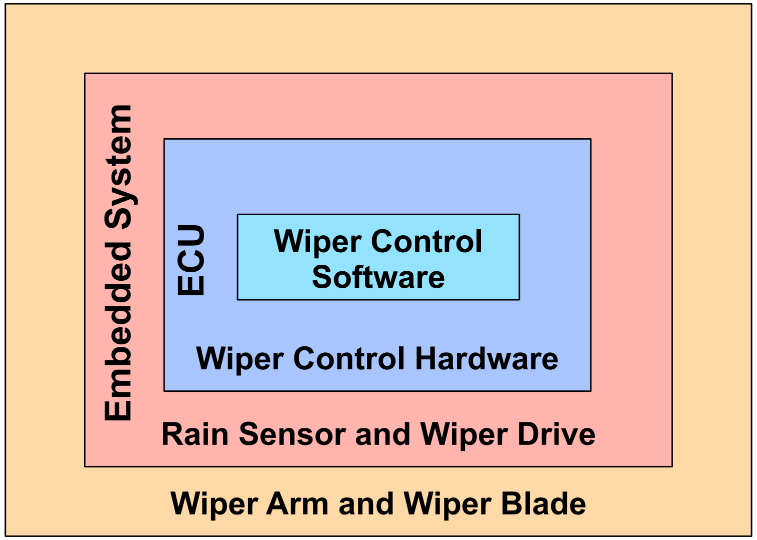


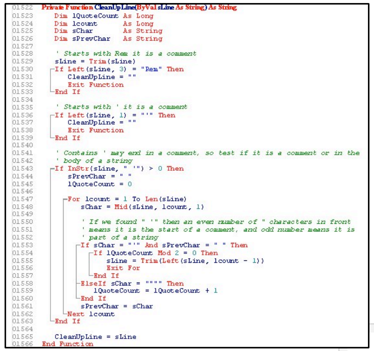
```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```



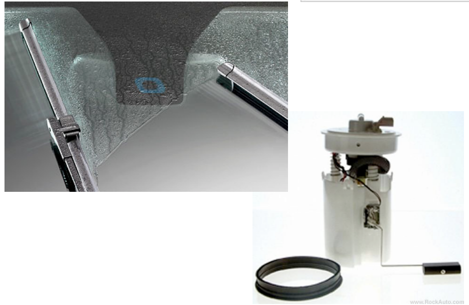
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS



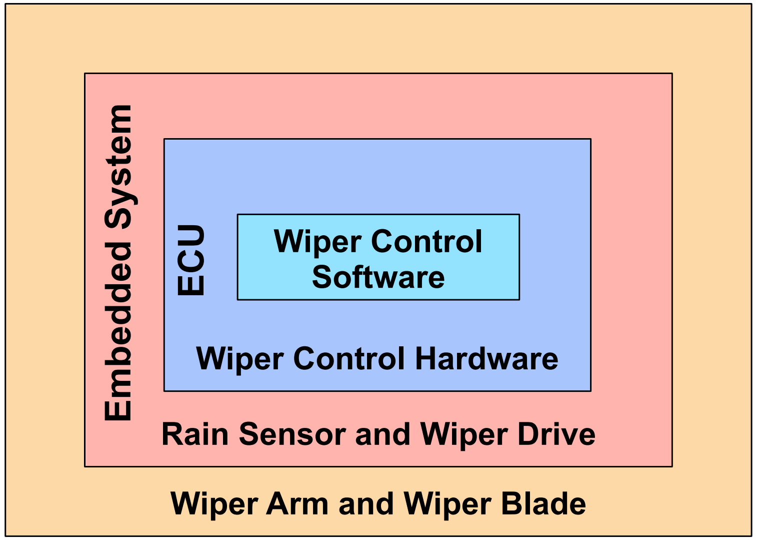


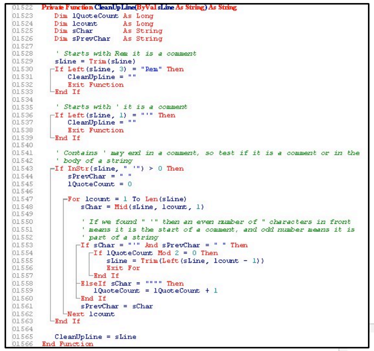
```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```



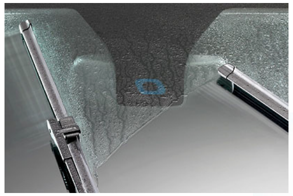
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS



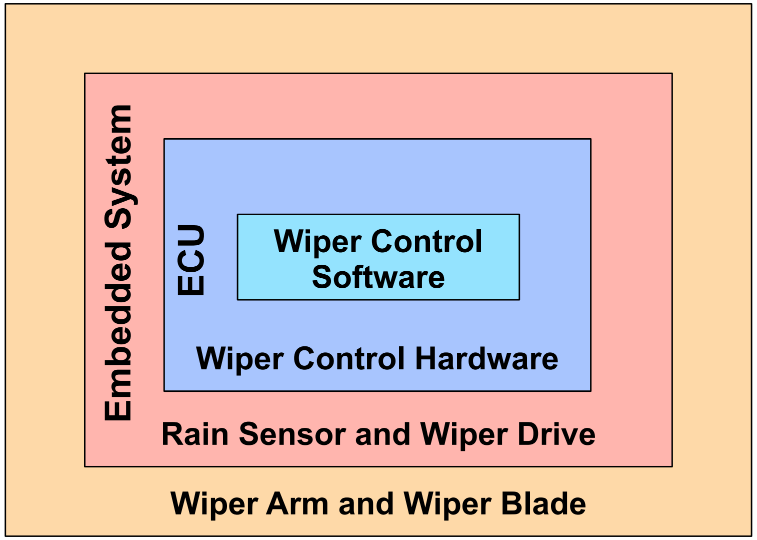


```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
   ┌If Left(sLine, 3) = "Rem" Then
   │    CleanUpLine = ""
   │    Exit Function
   └End If

    ' Starts with ' it is a comment
   ┌If Left(sLine, 1) = "'" Then
   │    CleanUpLine = ""
   │    Exit Function
   └End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
   ┌If InStr(sLine, " '") > 0 Then
   │    sPrevChar = " "
   │    lQuoteCount = 0
   │
   │   ┌For lcount = 1 To Len(sLine)
   │   │    sChar = Mid(sLine, lcount, 1)
   │   │
   │   │    ' If we found " '" then an even number of " characters in front
   │   │    ' means it is the start of a comment, and odd number means it is
   │   │    ' part of a string
   │   │   ┌If sChar = "'" And sPrevChar = " " Then
   │   │   │   ┌If lQuoteCount Mod 2 = 0 Then
   │   │   │   │    sLine = Trim(Left(sLine, lcount - 1))
   │   │   │   │    Exit For
   │   │   │   └End If
   │   │   ├ElseIf sChar = """" Then
   │   │   │    lQuoteCount = lQuoteCount + 1
   │   │   └End If
   │   │    sPrevChar = sChar
   │   └Next lcount
   └End If

    CleanUpLine = sLine
End Function
```



Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS



Wiper Control Software

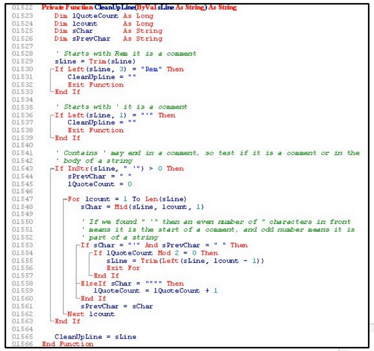
ECU

Wiper Control Hardware

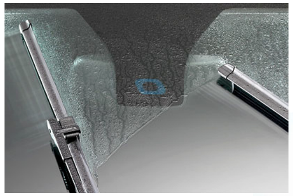
Embedded System

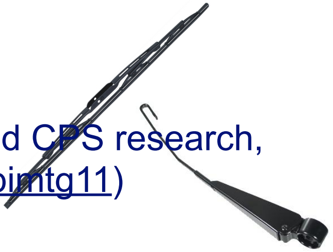
Rain Sensor and Wiper Drive

Wiper Arm and Wiper Blade



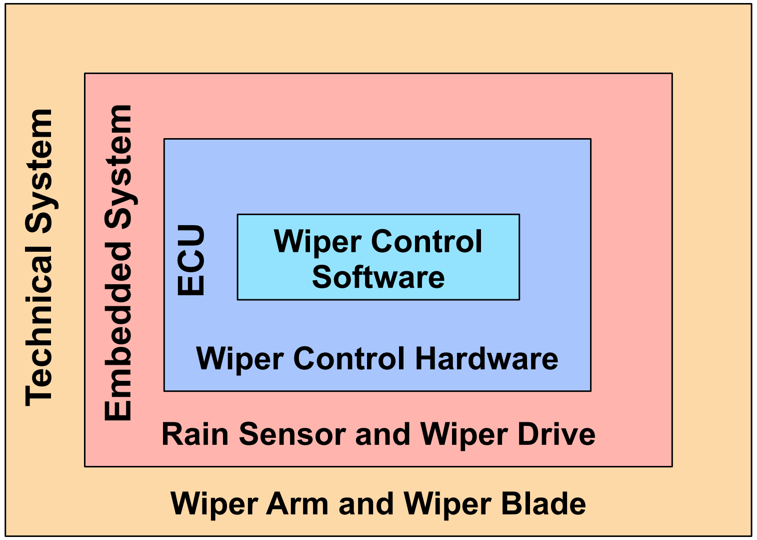
```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

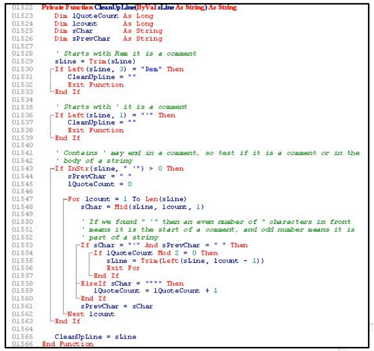




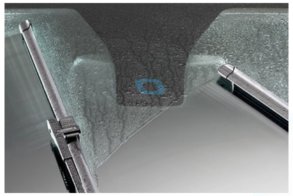
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS



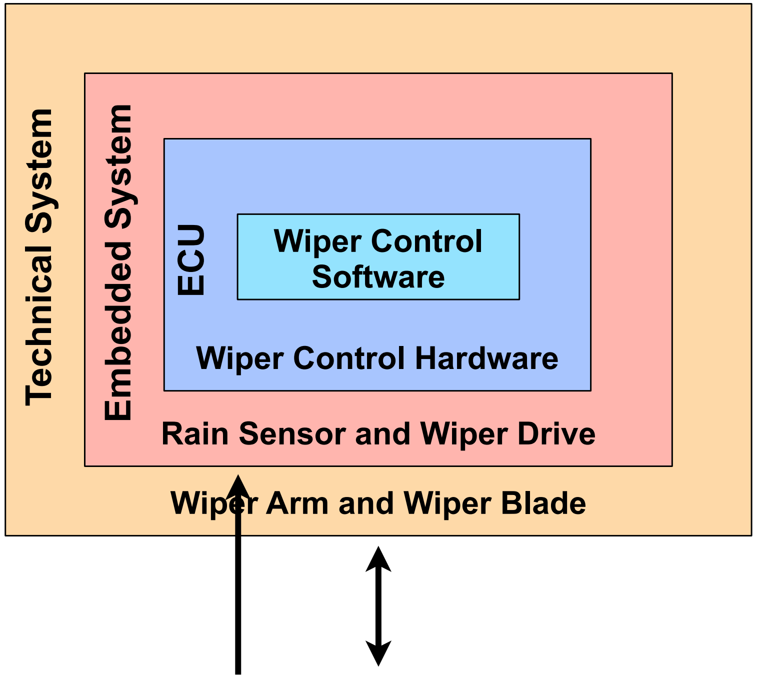


```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```



Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced):
# Example of a simple CPS



Technical System

Embedded System

ECU

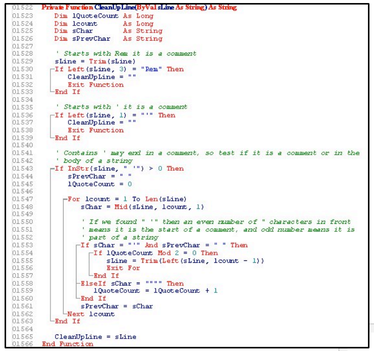
Wiper Control Software

Wiper Control Hardware

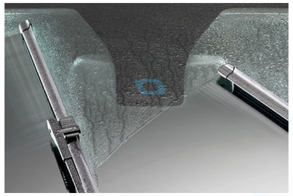
Rain Sensor and Wiper Drive

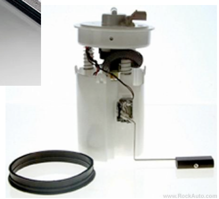
Wiper Arm and Wiper Blade

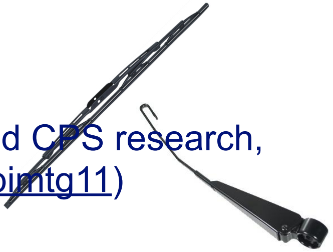
**Interaction with physical environment**



```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

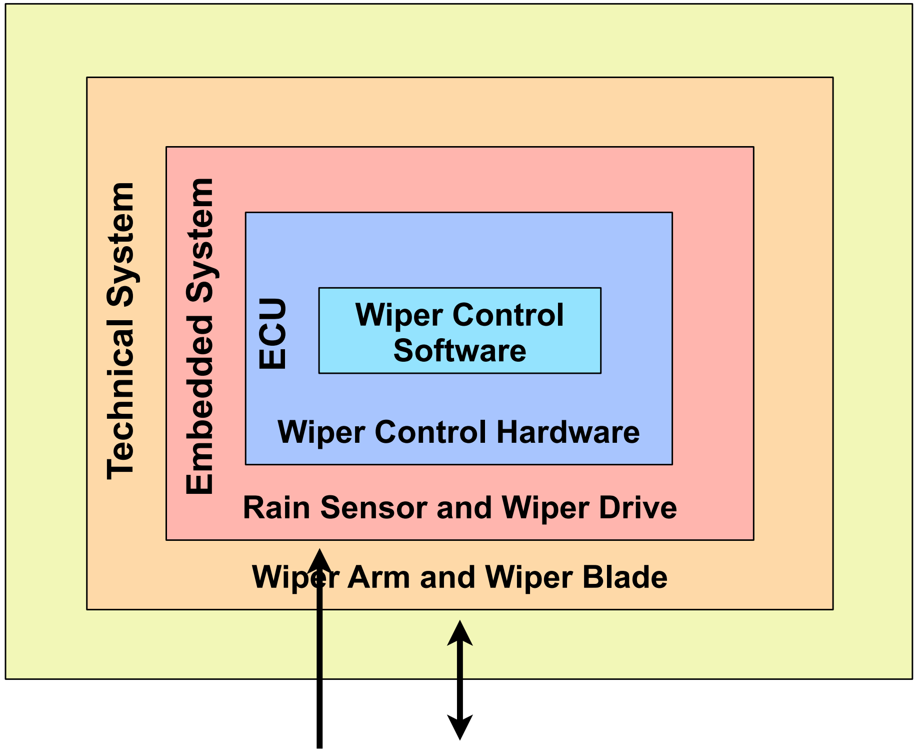




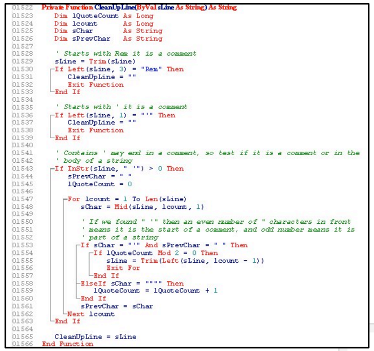


Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

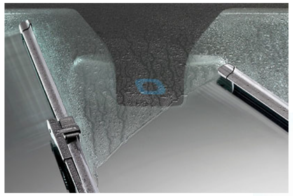
# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS

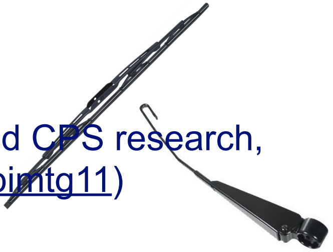


**Interaction with physical environment**



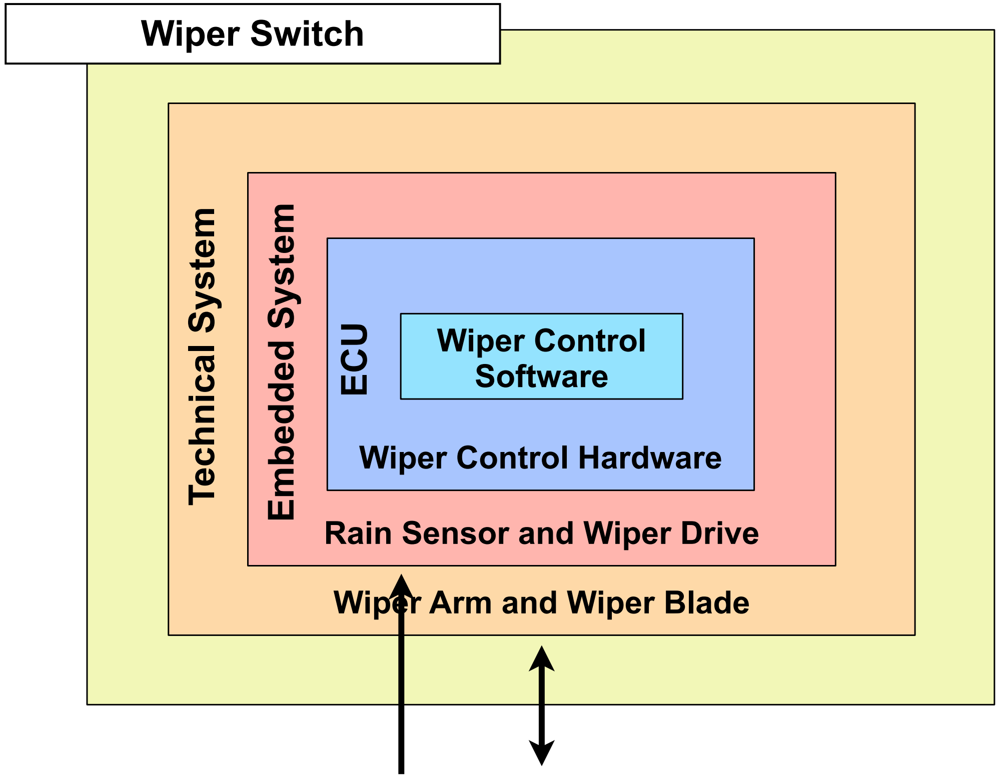
```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

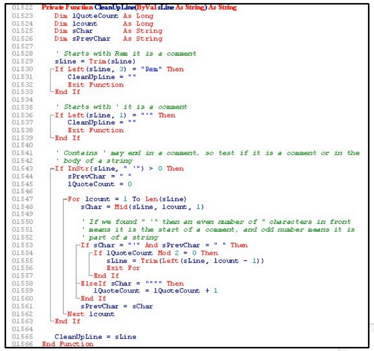


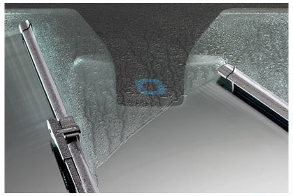


Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS

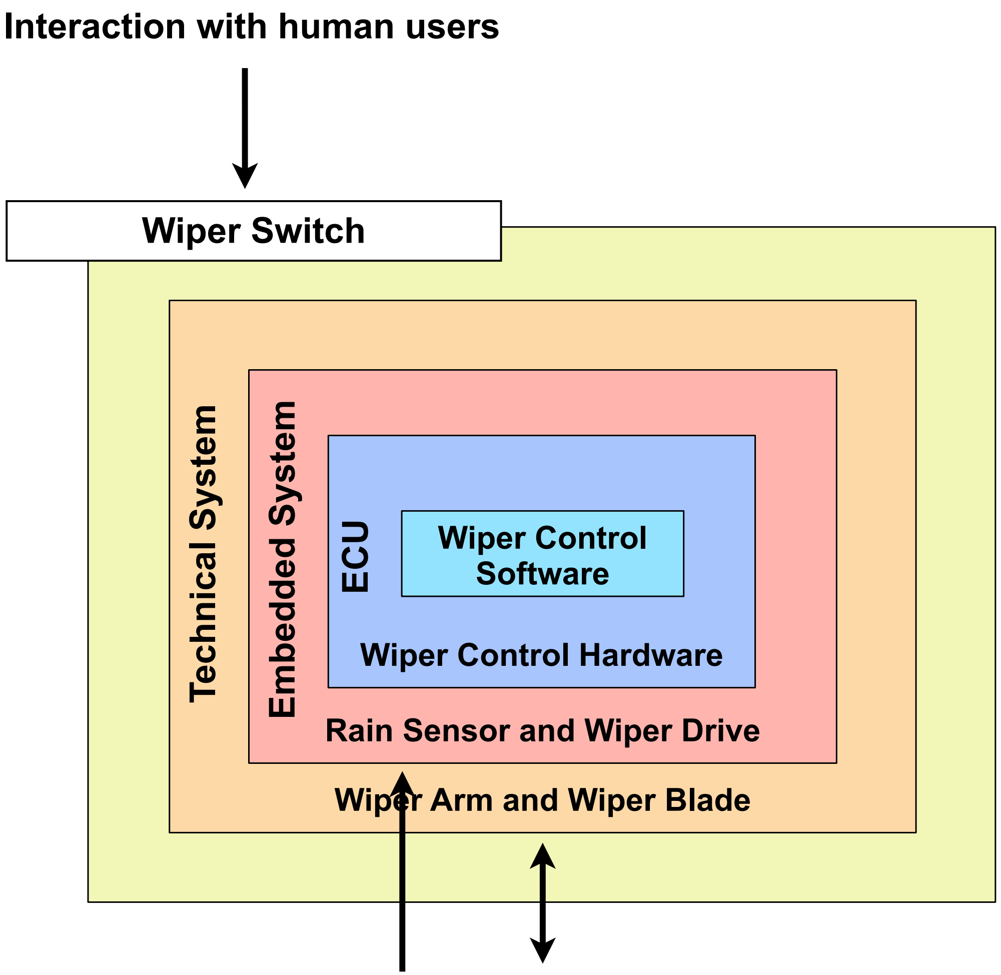


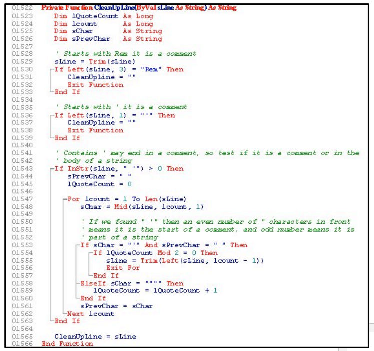

```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```



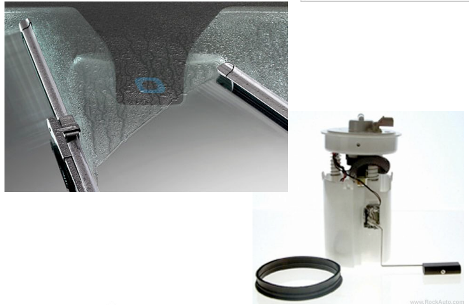
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS


Interaction with human users
Wiper Switch
Technical System
Embedded System
ECU
Wiper Control Software
Wiper Control Hardware
Rain Sensor and Wiper Drive
Wiper Arm and Wiper Blade
Interaction with physical environment

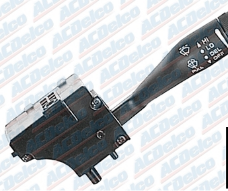


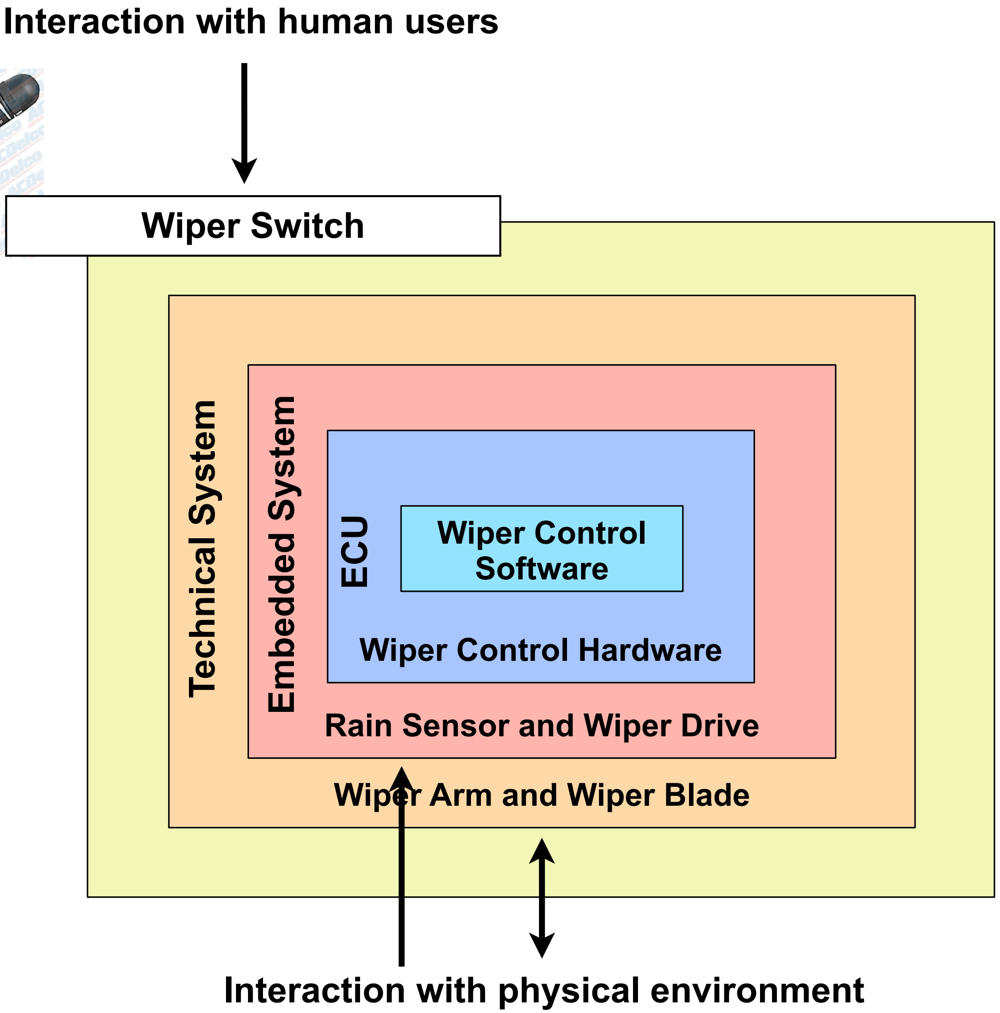
```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

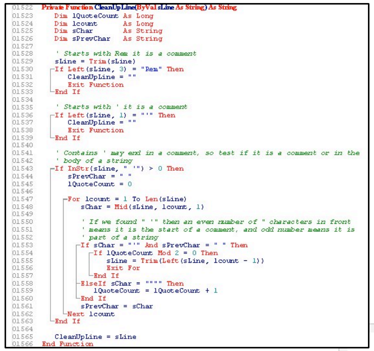


Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

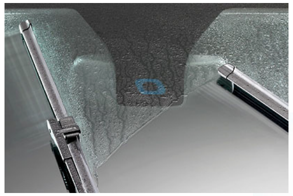
# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS




Interaction with human users
Wiper Switch
Technical System
Embedded System
ECU
Wiper Control Software
Wiper Control Hardware
Rain Sensor and Wiper Drive
Wiper Arm and Wiper Blade
Interaction with physical environment

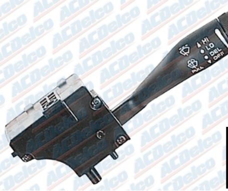


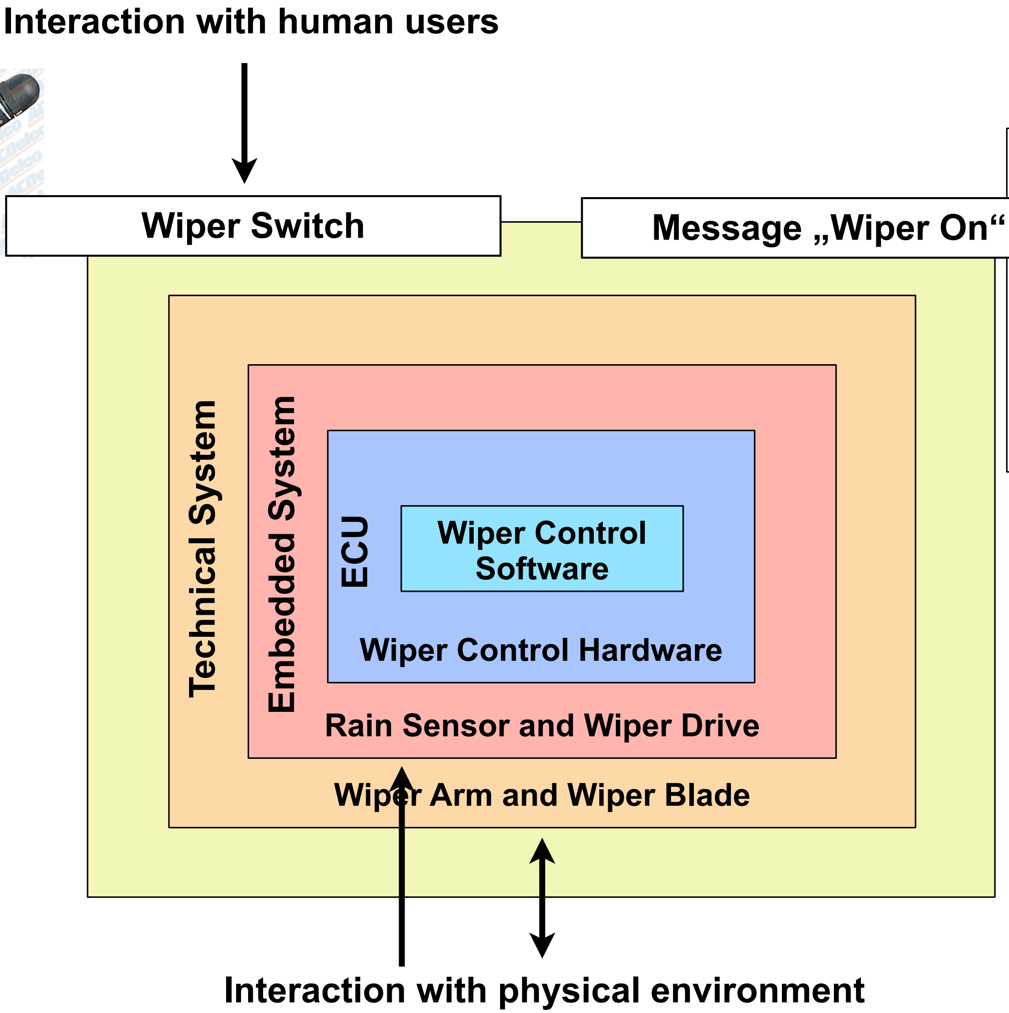
```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```



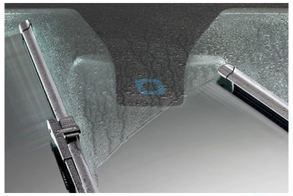
Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

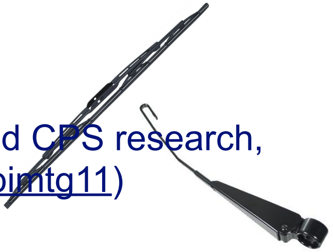
# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS




Interaction with human users
Wiper Switch
Message „Wiper On“
Technical System
Embedded System
ECU
Wiper Control Software
Wiper Control Hardware
Rain Sensor and Wiper Drive
Wiper Arm and Wiper Blade
Interaction with physical environment

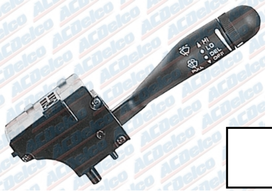
```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        eanUpLine = ""
        it Function
        f

        rts with ' it is a comment
        ft(sLine, 1) = "'" Then
        eanUpLine = ""
        it Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

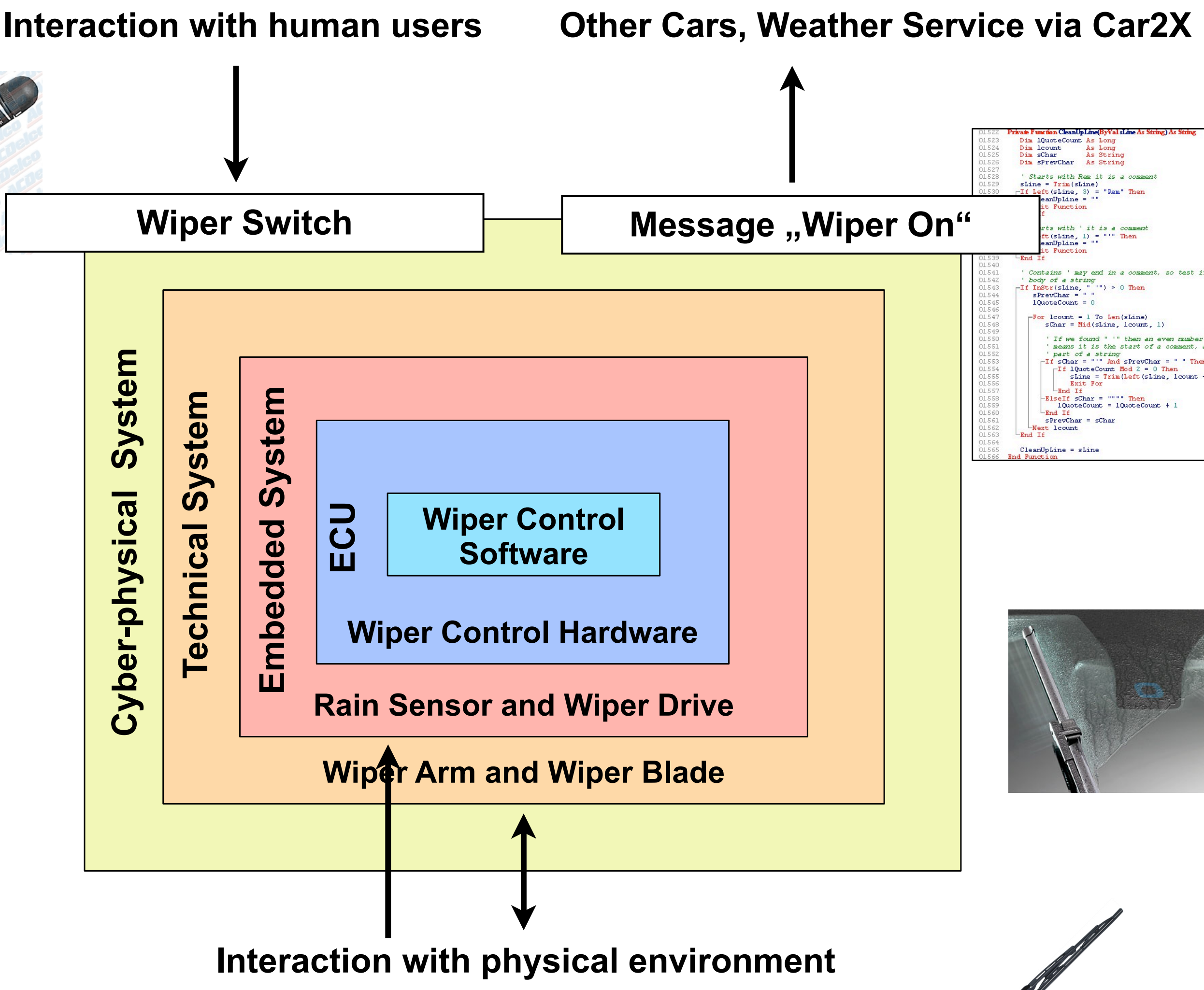


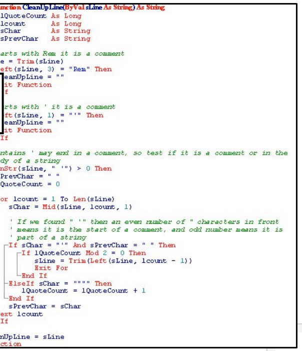


Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

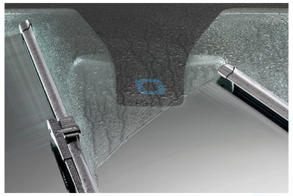
# Onion ring like structure of CPS (modified and enhanced): Example of a simple CPS

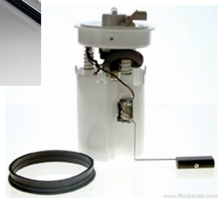


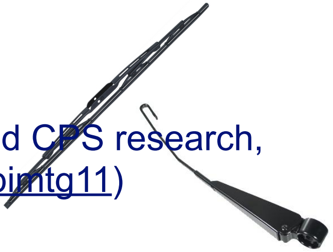

Interaction with human users
Other Cars, Weather Service via Car2X
Wiper Switch
Message „Wiper On“
Cyber-physical System
Technical System
Embedded System
ECU
Wiper Control Software
Wiper Control Hardware
Rain Sensor and Wiper Drive
Wiper Arm and Wiper Blade
Interaction with physical environment



```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
      eanUpLine = ""
      it Function
      f

      rts with ' it is a comment
      ft(sLine, 1) = "'" Then
      eanUpLine = ""
      it Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```







Source: Manfred Broy, Complex Systems Modeling and Engineering and CPS research, CPS Principal Investigator Meeting 2011 (http://www.cps-vo.org/group/pimtg11)

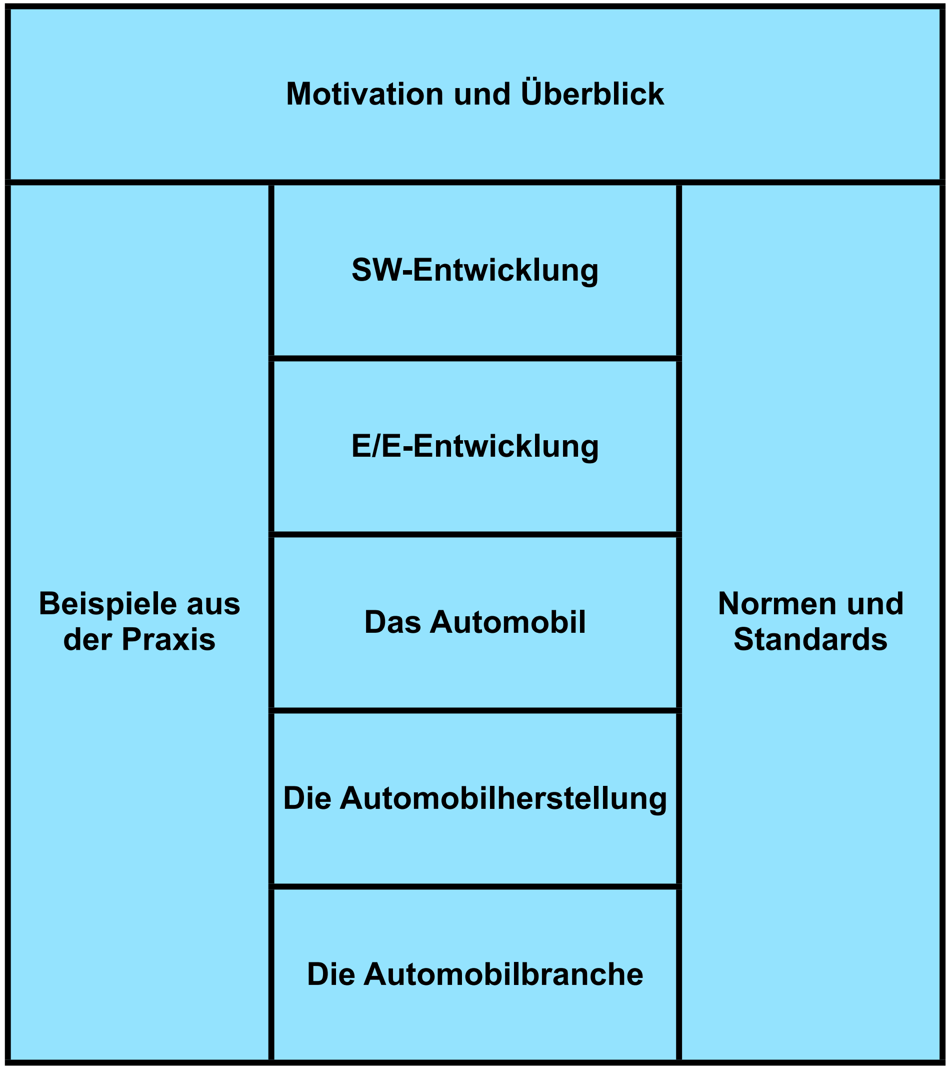
# Vorlesungen am Lehrstuhl für Fahrzeugmechatronik

http://tu-dresden.de/die_tu_dresden/fakultaeten/vkw/iad/professuren/fm/Studium/index_html

Weitere Informationen über
Prof. Dr.-Ing. Bernard Bäker
TU Dresden, Lehrstuhl für Fahrzeugmechatronik
http://tu-dresden.de/fzm

# Vorlesung Automotive Software Engineering
# Kurzbeschreibung

- Heutige Fahrzeuge haben teilweise mehr als 50 Steuergeräte, die weit über 500.000 Zeilen Code enthalten. Über bis zu vier verschiedene Kommunikationsbusse gehen hunderte von Nachrichten und tausende von Signalen. Über zwei Drittel aller Innovationen im Automobil sind schon heute software-basiert, ein Anstieg der Softwareentwicklungskosten an den gesamten Entwicklungskosten von derzeit ca. 4% auf über 10% wird prognostiziert. Ein Automobil bündelt so auf 5x2m viele Fragestellungen der Informatik, insbesondere der Entwicklung komplexer und zuverlässiger Softwaresysteme.
- Die Vorlesung führt in die Grundlagen und Besonderheiten des Software-Engineerings für elektronische Systeme im Automobil ein:
  - Verteilte und komplexe Systementwicklung zwischen OEM und Zulieferern
  - Sehr hohe Anforderungen an Zuverlässigkeit, Sicherheit (Safety und Security) sowie Echtzeitverhalten
  - Extreme Umweltbedingungen (mechanische Beanspruchung, Verbauraum, Temperatur, etc.)
  - Unterschiedliche Entwicklungs- und Lebenszyklen zwischen Produkt (Fahrzeug) und Software (Komponente)
  - Hoher Zeit- und Kostendruck mit vielen Änderungs- und Konfigurationsanforderungen
- Anhand zahlreicher Beispiele aus der Praxis werden die relevanten Entwicklungsprozesse und Modelle, Elektrik/Elektronik-Zielarchitekturen, Betriebssysteme und Bussysteme, Verfahren zum System- und Softwareentwurf inkl. der Unterstützungsprozesse und notwendiger Entwicklungswerkzeuge sowie relevante Normen und Standards vorgestellt.



Motivation und Überblick

Beispiele aus der Praxis

SW-Entwicklung

E/E-Entwicklung

Das Automobil

Die Automobilherstellung

Die Automobilbranche

Normen und Standards

# Lernziele
# Vorlesung Automotive Software Engineering

- Software Engineering im Anwendungskontext Automobil kennenlernen und verstehen
  - Automobilbranche als wichtiger Wirtschaftszweig
  - Automobilherstellung mit technischen und wirtschaftlichen Randbedingungen
  - Automobil als technisches System
  - E/E-Entwicklung zur Realisierung von Funktionen
  - SW-Entwicklung als Teil der E/E-Entwicklung
- Software als Mittel zur Realisierung von technischen Funktionen begreifen
- Relevante Normen und Standards beispielhaft kennenlernen
- Das Gelernte an Beispielen aus der Praxis reflektieren und wieder erkennen

# Vorlesung Automotive Software Engineering
## Inhalt und Zeitplanung

| | |
|---|---|
| 1. Motivation und Überblick | 23. April |
| 2. Die Automobilbranche | 23. April |
| 3. Die Automobilherstellung | 7. Mai |
| 4. Das Automobil | 7. Mai |
| 5. E/E-Entwicklung | 21. Mai |
| 6. SW-Entwicklung | 18. Juni |
| 7. Normen und Standards | 25. Juni |
| 8. Beispiele aus der Praxis | 9. Juli |
| Prüfung | tbd, Kernprüfungszeit bis Sa, 11.08.2012 |
| Exkursion Gläserne Manufaktur<br>Montage des VW-Phaeton<br>Dresden, Strassburger Platz | tbd, Dienstag nach Vorlesung |

# Unterlagen

- Unterlagen werden nach den Vorlesungen ins Netz gestellt
- http://st.inf.tu-dresden.de/content/index.php?node=teaching
- Dateiname enthält Name des Vorlesungsteils
- Beispiel:
  Vorlesungsteil: 1. Motivation und Überblick
  Dateiname: 01 ASE SS 2012 Motivation und Überblick

# Bernhard Hohlfeld: Ausbildung und Beruf

- 1973 - 1980
  Studium der Mathematik in Karlsruhe und Grenoble
- 1988
  Promotion in Informatik an der TU Kaiserslautern
- 2011
  Ernennung zum Honorarprofesor an der TU Dresden, Fakultät Informatik
- 1980 - 1990
  Wissenschaftlicher Mitarbeiter im Forschungszentrum Ulm der Daimler AG (ehemals Telefunken Forschungsinstitut Ulm). Arbeitsgebiete: Software Engineering, Programmverifikation, Programmiersprachen, IT-Sicherheit sowie Künstliche Intelligenz und Expertensysteme
- 1991 - 2008
  Abteilungsleiter in der Forschung der Daimler AG in den Bereichen Telematik und Softwaretechnologie; Aufgabengebiete: Mobilkommunikation für Fertigung und Verkehrstelematik, Multimedia-Kommunikation, Telematikdienste, Software-Architekturen, Software Reengineering, Software Wiederverwendung
- 2009 -
  Mitarbeiter der ICS AG; Arbeitsgebiete: Software-Engineering eingebetter Systeme, insbesondere Automotive Software Engineering, Funktionale Sicherheit, Vorgehensmodelle,

# Bernhard Hohlfeld: Engagement im Fachgebiet (Auswahl)

- Ca. 10 abgeschlossene Promotionen von Mitarbeitern, u.a.
  - Dr. Gabriel Schwefer (Prof. Ilka Phillipow, TU Ilmenau)
  - Prof. Dr. Jens Weiland (Prof. Ulrich Eisenecker, Universität Leipzig)
  - Dr. Walter Franz (Prof. Paul Kühn, Universität Stuttgart)
  - Dr. Reinhold Eberhardt (Prof. Ralf Steinmetz, TU Darmstadt)
  - Dr. Ralf Hinz (Prof. Alexander Schill, TU Dresden)
  - Dr. Albert Held (Prof. Alexander Schill, TU Dresden)
- Lehrbeauftragter an der Fakultät Informatik der TU Dresden und in den Fachbereichen Elektrotechnik und Informationstechnik sowie Informatik der TU Darmstadt
- 2005 - 2006
  Mitglied der Berufungskommission für die Professur „Eingebettete Systeme / Echtzeitsysteme“ an der Fakultät für Informatik der Universität Ulm
- 2004 - 2008
  Mitglied im Beirat des Kompetenzfeldes Informationstechnik im Verein Deutscher Ingenieure (VDI)

# Bernhard Hohlfeld: Engagement im Fachgebiet (Auswahl)

- 1997 bis 2003: Mitglied des Programmauschusses der GI/ITG Fachtagung „Kommunikation in Verteilten Systemen“ und des Leitungsgremiums der gleichnamigen Fachgruppe
- 2003 - heute
  Mitglied im Programmkomitee des Workshops „Automotive Software Engineering“ im Rahmen der Jahrestagungen der Gesellschaft für Informatik bzw. der Fachtagung SOFTWARE ENGINEERING
  2004 und 2008 Organisation und Leitung des Workshops
- 2005 - 2008
  Mitbegründung und Organisation der GI-Fachgruppe „Automotive Software Engineering“
  http://www.fg-ase.gi-ev.de/
- 2006
  Programmkomitee der Jahrestagung der Gesellschaft für Informatik in Dresden

# Bernhard Hohlfeld: Bezug zur TU Dresden

- Vorlesung Automotive Software Engineering
- Langjährige Zusammenarbeit mit Prof. Alexander Schill
  - Promotionen Dr. Albert Held und Dr. Ralf Hinz
  - Studien- und Diplomarbeiten
  - Projekte (Mobile Agenten, Mobile Computing)
- OUTPUT - Absolventenmesse der Fakultät Informatik
- Prof. Bernard Bäker, Fakultät Verkehrswissenschaften  
  Institut für Automobiltechnik (IAD), Lehrstuhl für Fahrzeugmechatronik  
  Ehemaliger Kollege (Daimler AG)
- KontaktMesse Verkehr - Absolventenmesse der Fakultät Verkehrswissenschaften
- Prof. Gerhard Fettweis, Lehrstuhl Mobile Nachrichtensysteme  
  Projekte „Breitbandmobilfunk“
- Beiträge auf Konferenzen
- Gesellschaft von Freunden und Förderern der Technischen Universität Dresden e. V. (GFF)

# ICS AG - Allgemeine Informationen

- Gesellschaftsform Aktiengesellschaft
- Stammkapital 2,4 Mio. EUR

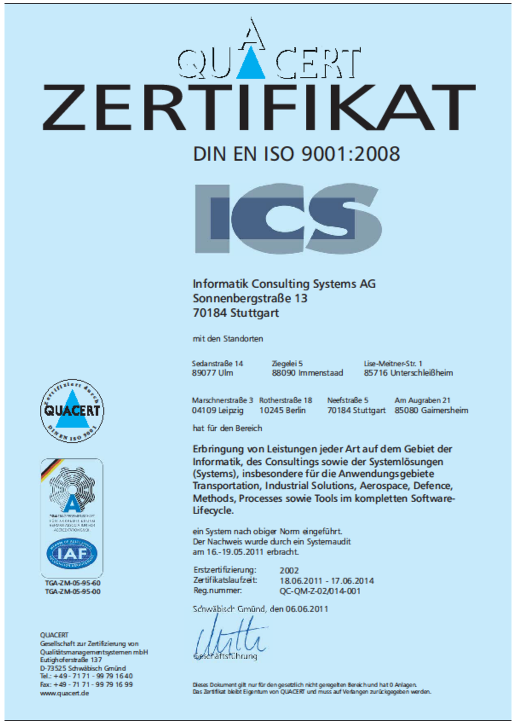
- Vorstand: Franz-Josef Winkel, Cid Kiefer

- Hauptsitz Stuttgart
- Geschäftsstellen Berlin, München, Leipzig, Immenstaad, Ulm, Braunschweig , Ingolstadt und Rüsselsheim

- Handelsregister
  Amtsgericht Stuttgart HRB Nr.: 18569
- UST. Id. Nr.: DE 147802488

- 150 Mitarbeiter, davon 120 produktive Ingenieure

- www.ics-ag.de



QUACERT

ZERTIFIKAT

DIN EN ISO 9001:2008

Informatik Consulting Systems AG
Sonnenbergstraße 13
70184 Stuttgart

mit den Standorten

Sedanstraße 14
89077 Ulm

Ziegelei 5
88090 Immenstaad

Lise-Meitner-Str. 1
85716 Unterschleißheim

Marschnerstraße 3
04109 Leipzig

Rotherstraße 18
10245 Berlin

Neefstraße 5
70184 Stuttgart

Am Augraben 21
85080 Gaimersheim

hat für den Bereich

**Erbringung von Leistungen jeder Art auf dem Gebiet der Informatik, des Consultings sowie der Systemlösungen (Systems), insbesondere für die Anwendungsgebiete Transportation, Industrial Solutions, Aerospace, Defence, Methods, Processes sowie Tools im kompletten Software-Lifecycle.**

ein System nach obiger Norm eingeführt.
Der Nachweis wurde durch ein Systemaudit
am 16.-19.05.2011 erbracht.

| | |
|---|---|
| Erstzertifizierung: | 2002 |
| Zertifikatslaufzeit: | 18.06.2011 - 17.06.2014 |
| Reg.nummer: | QC-QM-Z-02/014-001 |

Schwäbisch Gmünd, den 06.06.2011

Geschäftsführung

TGA-ZM-05-95-60
TGA-ZM-05-95-00

QUACERT
Gesellschaft zur Zertifizierung von
Qualitätsmanagementsystemen mbH
Eutighoferstraße 137
D-73525 Schwäbisch Gmünd
Tel.: +49 - 71 71 - 99 79 16 40
Fax: +49 - 71 71 - 99 79 16 99
www.quacert.de

Dieses Dokument gilt nur für den gesetzlich nicht geregelten Bereich und hat 0 Anlagen.
Das Zertifikat bleibt Eigentum von QUACERT und muss auf Verlangen zurückgegeben werden.

# ICS AG - Kunden

ICS

- Industrie
  - ABB
  - Alcatel SEL
  - Bombardier
  - CASSIDIAN
  - Daimler AG
  - Deuta-Werke
  - Diehl Aerospace
  - Dürr
  - Funkwerk IT
  - Hamburger Hochbahn
  - IBM
  - Infineon
  - Johnson Controls
  - Marconi
  - Philips
  - Robert Bosch
  - Rohde&Schwarz
  - RWE
  - Nortel Networks
  - Siemens Dematic
  - Siemens TS
  - Siemens VDO
  - Thales
  - Temic
  - T-Systems
  - Volkswagen
  - Vossloh IT
  - ZF Friedrichshafen
- Öffentlicher Bereich, Behörden
  - BWB
  - BKA
  - Universität Ulm
  - Universität Magdeburg
  - FH Weingarten
  - Universität Kassel
  - Stadt München
  - Stadt Frankfurt
  - Innenministerium BW
  - Staatsanwaltschaft Stuttgart

BOSCH

RHEINMETALL DEFENCE

SIEMENS

DAIMLER

BOMBARDIER

THALES

CASSIDIAN AN EADS COMPANY

DEUTA-WERKE

DIEHL Aerospace

# ICS AG - Kompetenzen

- System und Software Engineering für sicherheitskritische eingebettete Systeme in den Domänen
  - Eisenbahnwesen
  - Automobil
  - Luft- und Raumfahrt

- System und Software Engineering für Industrieautomatisierung, insbesondere Logistikprozesse

# Vorlesung

- **System und Software Engineering für sicherheitskritische eingebettete Systeme** in den Domänen
  - Eisenbahnwesen
  - **Automobil**
  - Luft- und Raumfahrt

- System und Software Engineering für Industrieautomatisierung, insbesondere Logistikprozesse

# Literatur

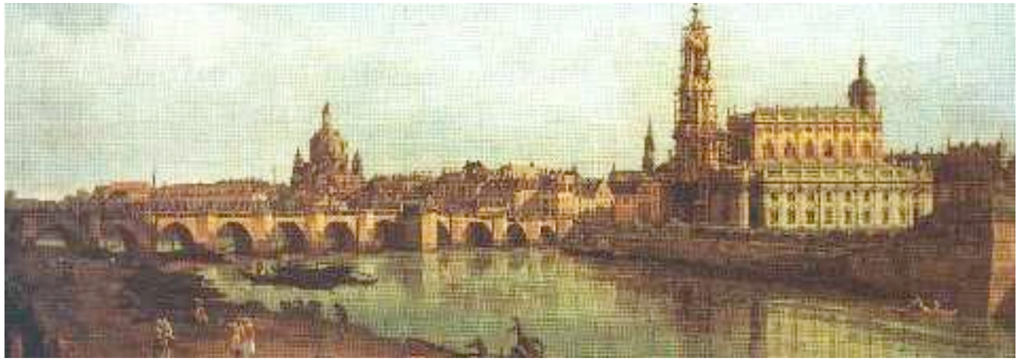
- J. Schäuffele,Th. Zurawka: Automotive SW Engineering, Vieweg+Teubner, 4. Auflage, 2010.
- P. Liggesmeyer, D. Rombach (Hrsg.): SW Engineering eingebetteter Systeme, Elsevier, 2005.
- O. Kindel, M. Friedrich: Softwareentwicklung mit AUTOSAR, dpunkt.verlag, 2009.
- AUTOSAR Software Modules, Specialized Glossary
  Von Vector Informatik GmbH zur Verfügung gestellt und an die Hörer verteilt.
- K. Reif: Automobilelektronik. Eine Einführung ..., Vieweg+Teubner, 3. Auflage, 2009.
- T. Trautmann: Grundlagen der Fahrzeugmechatronik, Vieweg+Teubner Verlag, 2009.
- M. Rausch: FlexRay - Grundlagen, Funktionsweise, Anwendung, Carl Hanser Verlag 2008.
- BOSCH: Kraftfahrtechnisches Taschenbuch, Vieweg+Teubner, 27. Auflage, 2011.
- Mercedes-Benz „Technik Transparent" Ausgabe 2009.
  Von Daimler AG zur Verfügung gestellt und an die Hörer verteilt. - wenn verfügbar
- http://www.volkswagen.de/de/Volkswagen/InnovationTechnik/technik-lexikon.html
- http://www.bmw.de/de/de/insights/technology/technology_guide/start.html
- http://www.kfztech.de/technik.htm
- Specialized Dictionary for Electronic Networks in Automotive and Automation Engineering, German - English - French - Japanese
  Von Vector Informatik GmbH zur Verfügung gestellt und an die Hörer verteilt.

# Noch Fragen?

# Noch Fragen?

Allwissend
bin ich nicht;
doch viel ist mir bewusst.
Goethes Faust 1, Vers 1582,
Mephistopheles

# Goethe in Dresden



Goethe bereiste Dresden mehrfach als Kunsttourist

„Dresden hat mir große Freude gemacht und meine Lust, an Kunst zu denken, wieder belebt. Es ist ein unglaublicher Schatz aller Art an diesem schönen Orte“, schrieb Johann Wolfgang von Goethe (1749-1832) nach einem seiner Besuche 1790 in Dresden begeistert. Insgesamt sieben Mal bereiste der Dichter die Stadt an der Elbe als einer der ersten und berühmtesten Kulturtouristen.

Als Goethe Dresden als 19jähriger Student zum ersten Mal besuchte, wohnte er bei dem Schuhmacher Johann Gottfried Haucke in der Dresdner Friedrichstadt. Eine Gedenktafel am Haus Friedrichstraße 5 erinnert an Goethe. Auch in der Umgebung sind noch zahlreiche Gebäude so erhalten, wie sie Goethe erlebte, so das barocke Marcolini-Palais, das heute als Krankenhaus genutzt wird, und in dessen Park der großartige Neptunbrunnen von 1744 einen Besuch lohnt.

"Die wenigen Tage meines Aufenthaltes in Dresden waren allein der Gemäldegalerie gewidmet" erinnert sich Goethe in seiner Autobiographie ,,Dichtung und Wahrheit". Mit Ungeduld erwartete er die Öffnung des ,,Heiligtums", das damals noch im Johanneum untergebracht war. Heute zeigt hier das Verkehrsmuseum Transportmittel aller Art vom Pferdefuhrwerk bis zum Auto.

Quelle http://www.dresden-congress.eu/Dresden-zur-Goethezeit.html

# Wo wurde dieses Fahrzeug aufgenommen?