A Cabot Technical Paper: CI Plus

This paper discusses what CI and CI Plus are, the reasons behind their creation and the implications for end-users, suppliers and customers. It then reflects on how Cabot set about developing, testing and completing its CI Plus implementation (Keystone) even before the hardware was available.

What are CI and CI Plus?

DVB Common Interface (CI) is a specification[1] that was published in 1997 to allow different devices to interoperate using a well defined protocol.

A content provider could sell a customer a hardware module (known as a CAM) that would plug into their television or set top box (Host), allowing them access to premium content. As long as both the CAM and Host conformed to the CI specification they would work together (Figure 1).

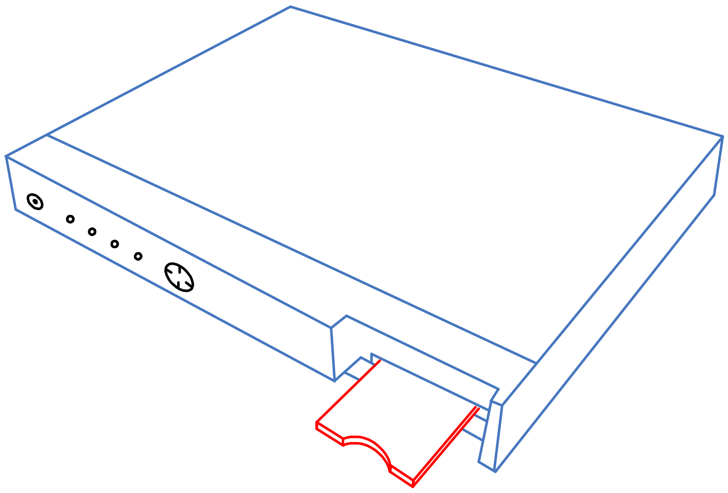


Figure 1: CI Host and CAM

This common standard benefited customers and content providers by ensuring that all devices would work together, preventing one manufacturer or provider from dominating. The CAMs provided basic user authentication with a smart card and/or pin meaning that only legitimate customers could decrypt the premium content.

CI was perfectly adequate when it was introduced, but worldwide surrounding media has changed dramatically since then. The Internet has opened new opportunities for content providers but has also brought with it new problems. The availability of high speed internet connections and the ease with which data can be shared online has caused a lot of headaches in both the video and music industry. Protecting the ownership of content has never been more important.

[1] Defined in EN 50221-1997

Consequently, CI no longer provides the protection required by today's content providers. As a result, in summer 2007 a number of consumer electronics manufacturers[2] created the CI+ Forum. Their goal was to extend the existing CI standard to provide greater security and richer functionality in order to meet the demands of both content providers and end users.

The first CI+ specification (v1.00) was published in January 2008 and then in November 2008 the CI+ Forum was disbanded and the CI Plus Limited Liability Partnership (CI Plus LLP) was formed. In early 2009 the LLP released the CI Plus 1.1 specification and shortly after 1.2.

Content Protection

Hundreds of millions of pounds is invested in creating high quality content every year and the potential loss of revenue if that content is not protected from copying can be enormous. With the availability and distribution of High Definition material expanding at an impressive rate the need to protect that content has become increasingly important.

The original CI system suffered a fatal flaw in that content that was decrypted by the CAM was sent unencrypted over the PCMCIA interface. This meant that anyone with the inclination could eavesdrop the decrypted data, either for personal use, or potentially to be uploaded to the internet and shared online or sold without any copy protection.

CI Plus addresses this problem in two ways, firstly by ensuring that content is not sent decrypted across the PCMCIA interface and secondly by making sure that both the Host and the CAM are authenticated. The authentication system uses a certificate hierarchy ensuring that only Hosts and CAMs that have been issued certificates from the official CI Plus LLP will pass authentication.

The protection of content over the PCMCIA interface is handled by using a diffie-hellman key exchange. The CAM and Host generate a private key to be used to encrypt data traversing the venerable PCMCIA bus. The very nature of the diffie-hellman exchange prevents a third party from obtaining the private key required to decrypt the content. The key used for encryption is also cycled frequently, meaning that, even if it was compromised, the amount of content that could be decrypted would be negligible (a few seconds).

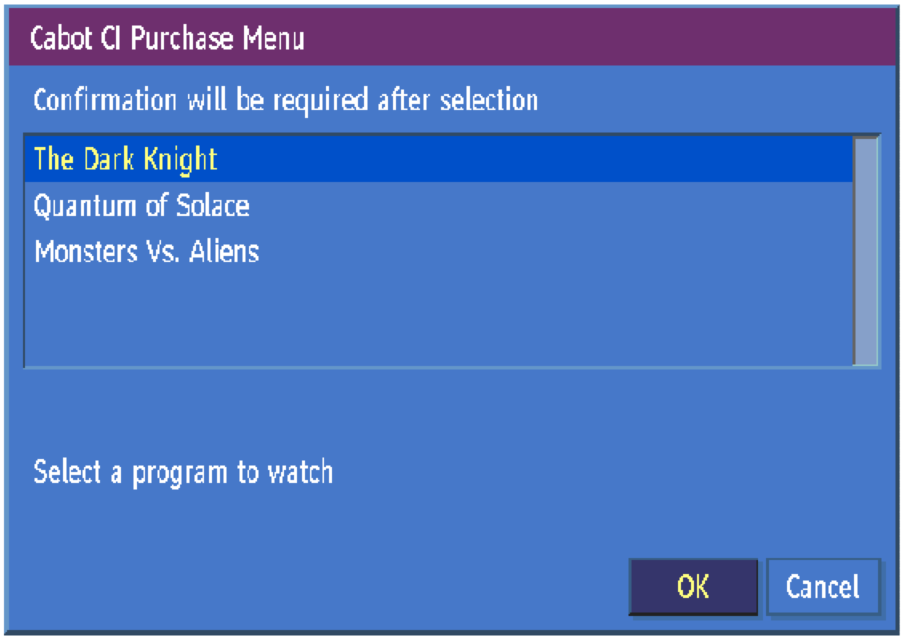
CI Plus also offers content providers more control over how their content is used. It has a concept of Usage Rights Information (URI) that allows a provider to enforce how a host outputs protected content. For example, HD video can be restricted to only being played out over HDMI with HDCP enabled or down-scaled when displayed over an analogue connection. A provider can specify the duration that content can be stored on a host and how many copies are allowed to be produced. There is also support for revocation lists, both of CI Plus hardware and HDCP compatible devices, ensuring compromised equipment can be prevented from accessing protected content.

User Experience

The original CI specification provides a functional but very limited user experience consisting exclusively of a pure text based menu and a data entry prompt (Figure

[2] Neotion, Panasonic, Philips, Samsung, SmarDTV and Sony

2). The CAM/content provider could only edit the text in these user interface elements, everything else is decided by the host implementer, this meant that the interface would look different depending on the host and it made it difficult to differentiate one CAM from another.


Cabot CI Purchase Menu

Confirmation will be required after selection

The Dark Knight
Quantum of Solace
Monsters Vs. Aliens

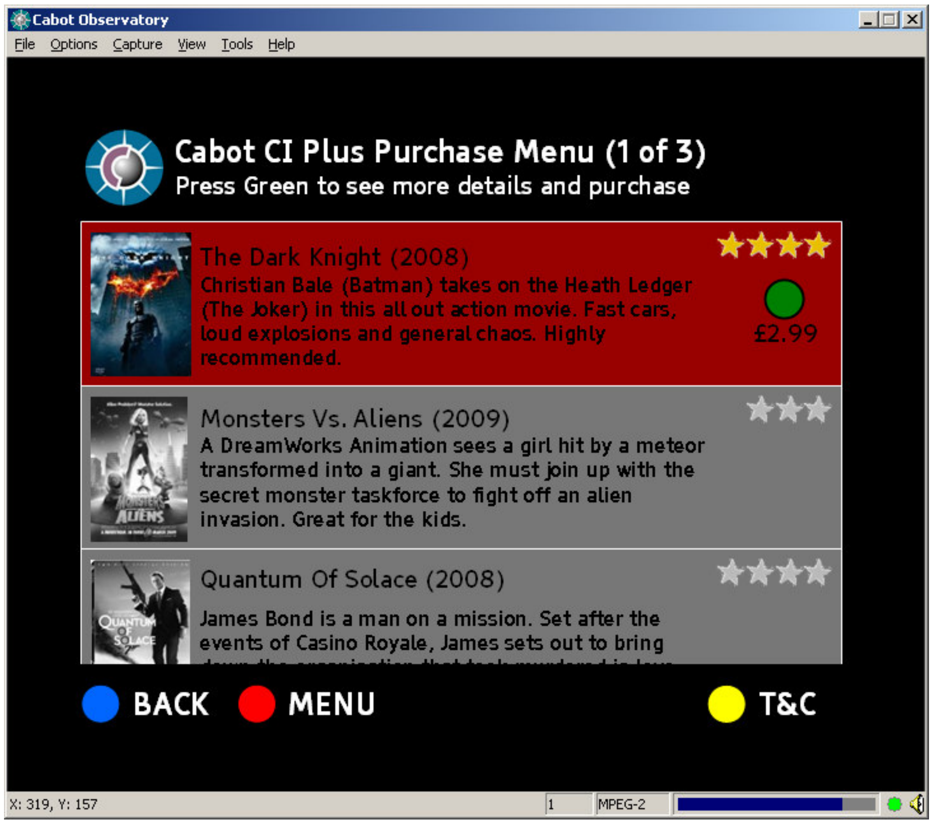
Select a program to watch

Figure 2: Traditional CI Menu

CI Plus offers a rich user experience by allowing MHEG applications to be downloaded and run directly from a CAM. This lets a CAM manufacturer or content provider display complex user interface elements at arbitrary positions such as text, buttons, images and vectors. Since the entire interface is provided by the CAM it will appear the same across all hosts.

More important than this, it allows content providers to brand their user experience (for example with logos) and differentiate them from other providers. It also allows users to benefit from a high quality dynamic user interface that is tailored to the content/CAM provider's offering.

An example of what can be achieved using the CI Plus MHEG applications can be seen in
Figure *3*.



Cabot CI Plus Purchase Menu (1 of 3)
Press Green to see more details and purchase

The Dark Knight (2008)
Christian Bale (Batman) takes on the Heath Ledger (The Joker) in this all out action movie. Fast cars, loud explosions and general chaos. Highly recommended.
£2.99

Monsters Vs. Aliens (2009)
A DreamWorks Animation sees a girl hit by a meteor transformed into a giant. She must join up with the secret monster taskforce to fight off an alien invasion. Great for the kids.

Quantum Of Solace (2008)
James Bond is a man on a mission. Set after the events of Casino Royale, James sets out to bring

Figure 3: Example CI Plus MHEG Application

Implementing CI Plus

How do you develop a product to one half of a specification when you don't have an implementation of the other half? When CI Plus was first announced Cabot realised that the CAM manufactures would not have hardware available for testing for at least a year. As a result, Cabot decided to write a CI Plus CAM emulator. This would serve two main purposes, firstly to allow us to start implementing CI Plus support in our host stack and secondly to allow automated testing of both CI and CI Plus functionality even after completing the work.

With these requirements in mind it was decided that the emulator be written in the Python programming language. This would allow us to integrate it with our automated test suite (Robotester), allowing both new and old CI functionality to be tested automatically. This autonomous testing ensures that any bugs caused by changes to our CI Plus implementation are flagged up and can be rectified quickly.

At first the emulator connected to Cabot's Observatory system (running on Windows) via a TCP/IP socket connection. This took the place of the PCMCIA interface that the real CAM uses. The socket implementation was handled at a low level (ICE Layer) which meant that the CI Plus stack wouldn't know the difference between data being sent and received over PCMCIA or TCP/IP (Figure 4). This decision was taken to ensure that as much of the real system is exercised by the Python emulator as possible.

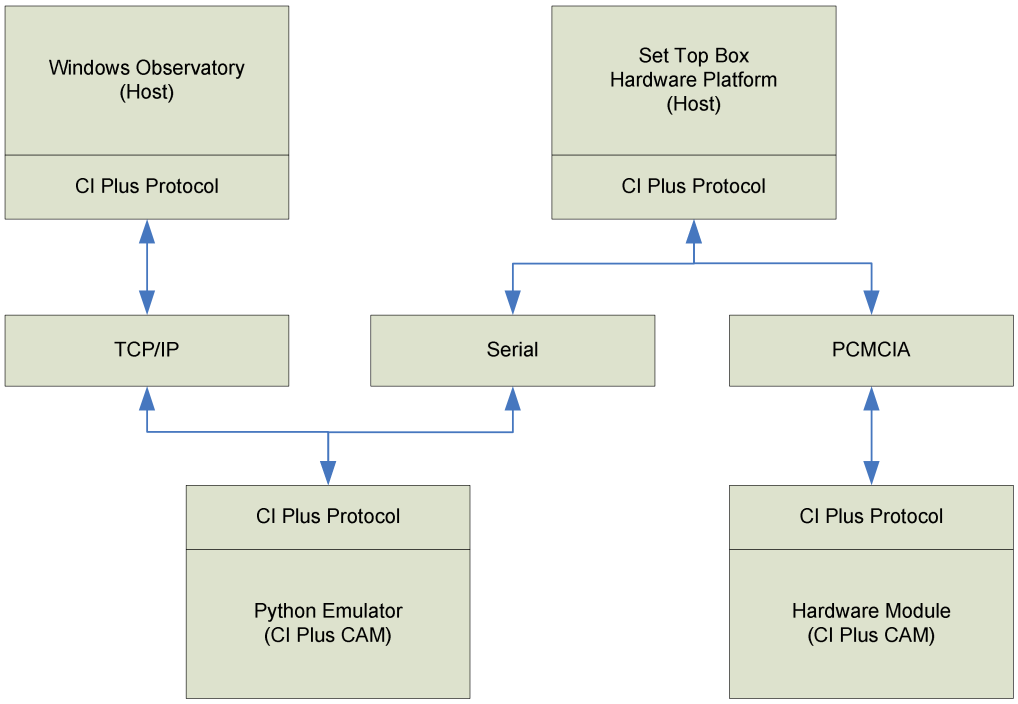
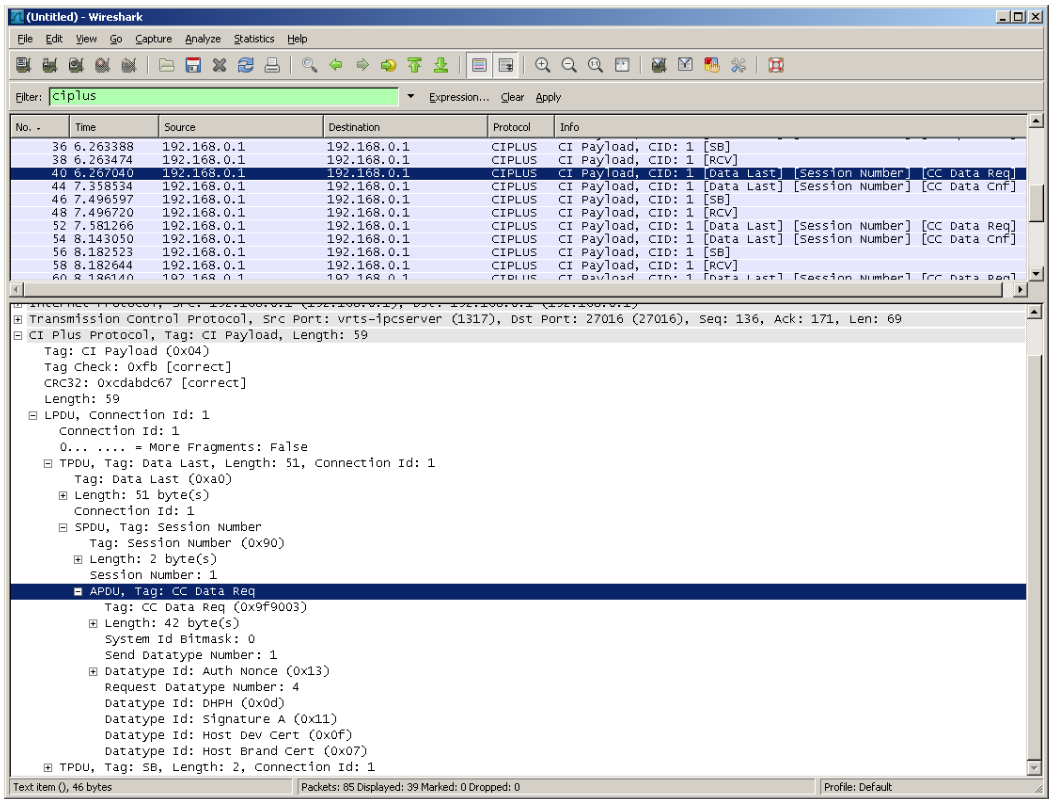

Windows Observatory (Host)
CI Plus Protocol
Set Top Box Hardware Platform (Host)
CI Plus Protocol
TCP/IP
Serial
PCMCIA
CI Plus Protocol
Python Emulator (CI Plus CAM)
CI Plus Protocol
Hardware Module (CI Plus CAM)

Figure 4: CI Plus Connectivity

The emulator was written by a separate team of engineers to those working on the host implementation. This ensured that any discrepancies due to interpretation of the specification could be identified early on and resolved either internally or with the CI+ Forum/LLP. This was especially important with regard to the cryptographic algorithms and processing of certificate data where even the slightest difference would cause a failure.

During development it became clear that a way of analysing the CI Plus packets being exchanged with the Python emulator was required. Since the communication was handled over a TCP/IP connection it was decided that a Wireshark[3] plug-in could provide the required functionality. Wireshark is an extensible open source network traffic analysis tool which provided the microscope required to analyse the CI Plus protocol either in real-time or post-mortem. It also provided another independent implementation of the CI Plus specification; the plug-in comprehensively understands both CI and CI Plus (Figure *5*), flagging any corruption or malformed packets.

[3] www.wireshark.org


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: ciplus
Expression... Clear Apply

```
No.  Time       Source        Destination   Protocol  Info
  36 6.263388   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [SB]
  38 6.263474   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [RCV]
  40 6.267040   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [Data Last] [Session Number] [CC Data Req]
  44 7.358534   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [Data Last] [Session Number] [CC Data Cnf]
  46 7.496597   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [SB]
  48 7.496720   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [RCV]
  52 7.581266   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [Data Last] [Session Number] [CC Data Req]
  54 8.143050   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [Data Last] [Session Number] [CC Data Cnf]
  56 8.182523   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [SB]
  58 8.182644   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [RCV]
  60 8.186140   192.168.0.1   192.168.0.1   CIPLUS    CI Payload, CID: 1 [Data Last] [Session Number] [CC Data Req]
```

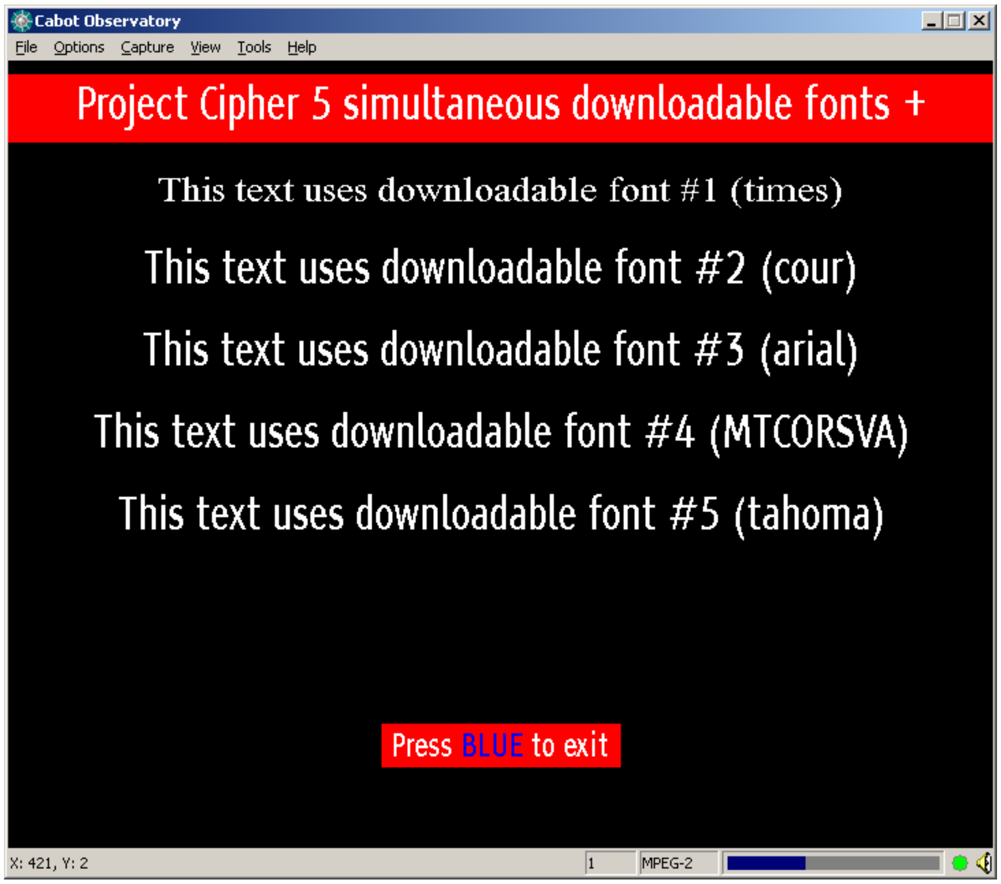
```
⊞ Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: vrts-ipcserver (1317), Dst Port: 27016 (27016), Seq: 136, Ack: 171, Len: 69
⊟ CI Plus Protocol, Tag: CI Payload, Length: 59
    Tag: CI Payload (0x04)
    Tag Check: 0xfb [correct]
    CRC32: 0xcdabdc67 [correct]
    Length: 59
  ⊟ LPDU, Connection Id: 1
      Connection Id: 1
      0... .... = More Fragments: False
    ⊟ TPDU, Tag: Data Last, Length: 51, Connection Id: 1
        Tag: Data Last (0xa0)
      ⊞ Length: 51 byte(s)
        Connection Id: 1
      ⊟ SPDU, Tag: Session Number
          Tag: Session Number (0x90)
        ⊞ Length: 2 byte(s)
          Session Number: 1
        ⊟ APDU, Tag: CC Data Req
            Tag: CC Data Req (0x9f9003)
          ⊞ Length: 42 byte(s)
            System Id Bitmask: 0
            Send Datatype Number: 1
          ⊞ Datatype Id: Auth Nonce (0x13)
            Request Datatype Number: 4
            Datatype Id: DHPH (0x0d)
            Datatype Id: Signature A (0x11)
            Datatype Id: Host Dev Cert (0x0f)
            Datatype Id: Host Brand Cert (0x07)
    ⊞ TPDU, Tag: SB, Length: 2, Connection Id: 1
```

```
Text item (), 46 bytes | Packets: 85 Displayed: 39 Marked: 0 Dropped: 0 | Profile: Default
```

Figure 5: Wireshark analysing CI Plus protocol

The emulator exercises CI Plus as well as regular CI functionality. For the MMI Cabot's MHEG test and design team wrote a number of CI Plus Profile MHEG applications, these test the extended functionality such as data pipes and downloadable fonts (
Figure 6). These MHEG applications are integrated into the Python emulator for automated testing and verification.

As the CI Plus solution matured, development moved onto a CI Plus capable hardware platform. At this time the Python emulator was modified to allow communication over a serial connection (again, replacing the PCMCIA layer). This allowed testing of all the functionality that had been implemented under Windows directly on the hardware platform. All this was taking place before the CAM manufacturers had even announced the availability of testing kits.



Project Cipher 5 simultaneous downloadable fonts +

This text uses downloadable font #1 (times)

This text uses downloadable font #2 (cour)

This text uses downloadable font #3 (arial)

This text uses downloadable font #4 (MTCORSVA)

This text uses downloadable font #5 (tahoma)

Press BLUE to exit

Figure 6: CI MHEG application demonstrating downloadable fonts.

Once Cabot obtained and began testing with the real hardware, it quickly became apparent that the emulator had proved to be a real success story. Within a matter of weeks and a few clarification e-mails later, Cabot's CI Plus implementation was fully working with real CI Plus CAMs!

Conclusion

Cabot's CI Plus stack (Keystone) has now been successfully tested against both Neotion and SmarDTV CI Plus CAMs, two of the founding members of the CI Plus LLP. The stack has also been independently verified using the Python emulator and has been tested to be backwards compatible with existing CI CAMs.

Keystone is available as an integrated part of Cabot's Aurora product and also as a standalone component that can be integrated into a third party receiver stack via a set of APIs called INCA.

If you would like any further information regarding the topic covered in this paper, please contact info@cabot.co.uk

CABOT

Cabot Communications Ltd, Verona House, Filwood Road, Bristol, BS16 3RY, UK
Tel: +44 (0) 117 958 4232 Fax: +44 (0) 117 958 4168 Email: info@cabot.co.uk