Design Document

Linking of Kiple's Online Payment System to Sigmatech's Maxpark Parking System.

V1.0
Revision E
Late February 2019

Revision History

#	Version	Description	By	Date
1.	V1.0	Initial Design Document	TSL	24 Aug 2016
2.	V1.0B	Added “$” and “T” Commands. General Clarifications.	TSL	31 Oct 2016
3.	V1.0C	Added “Loopback” data. . General Clarifications.	TSL	24 Nov 2016
4	V1.0D	Changes made after feedback from the System Trial Run.	TSL	07 Aug 2017
5.	V1.0E	Modified the “G” command to handle e-Tickets	TSL	21 Feb 2019

1. Overview

1.1 The purpose of this system is to link a Cloud-based Payment system to a Maxpark parking system and allow it to make Parking Fee payments for tickets generated by Maxpark.

1.2 The system structure shall be as the figure below:

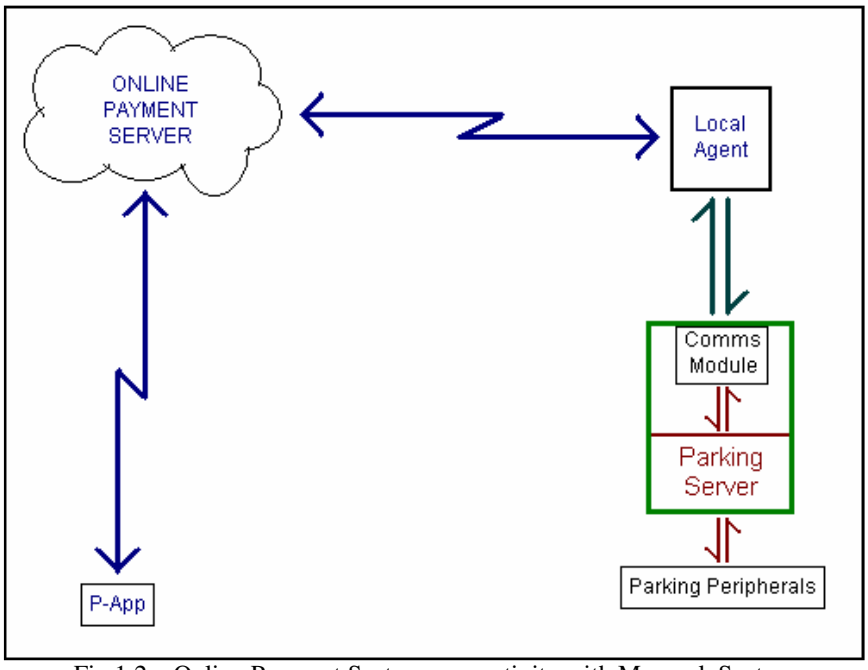

Fig 1.2 – Online Payment System connectivity with Maxpark System

1.2.1 P-App - This is an app installed on a Customer's smartphone. Customers will use this app to interact with the Online Payment System.

1.2.2 Online Payment Server – This is a server in the cloud that will perform the online payments

1.2.3 Local Agent – This is program installed on an internet enabled PC and will be installed at site. It will act as the bridge between the Online Payment System and the Parking Server.

1.2.4 Comms Module – This is an additional module in the Parking Server that will communicate with the Local Agent

1.2.5 Parking Server – Existing parking server.

1.2.6 Parking Peripherals – Existing Ticket Dispensers, Autopay Machines and other peripherals that are used in the parking operation.

1.3 P-App, Online Payment Server and Local Agent shall be supplied by Others.

1.4 The Comms Module, Parking Server and Parking Peripherals shall be supplied by Sigma

2. Expected Operational Flow

2.1 The standard system operational flow shall be as follows

2.2 Customers will take a ticket on arrival at the parking site as usual

2.3 Ticket shall contain a Standard 1D Barcode plus a 2D barcode (QR-Code format)

Figure 2.3 – Proposed Ticket

2.4 The Standard 1D Barcode shall be used for non-online payment only
2.5 The QR-Code shall be in the following format: SITE_CODE ; TICKET_DATA where
2.5.1 SITE_CODE = 4 digit site code that shall be unique for each of Sigma's parking sites.
2.5.2 ; = Semicolon used as separator between site code and ticket data
2.5.3 TICKET_DATA = Encrypted ticketing data (20-digits), same as found in the Standard 1D Barcode
2.6 To pay the Parking Fee, Customers will scan the QR-Code and the P-App will send the barcode to the Online Payment Server.
2.7 When the Online Payment System receives this data, it shall contact the appropriate Local Agent based on the contents of the SITE_CODE.
2.8 Local Agent will send TICKET_DATA to the Comms Module to determine the Parking Fee.
2.9 The Parking Fee will then be displayed on the Customers' smartphone by the P-App.
2.10 Customer can then "Authorize Payment" and the Online Payment Server will send an "Authorization Code" to the Parking System.
2.11 The Parking System shall apply the payment and return a "Receipt Number".
2.12 Online Payment Server will send the receipt number to the P-App.
2.13 Customer can then exit the parking site by inserting the ticket at the exit validator.

2.14 Notes

2.14.1 Tickets paid Online are still subject to the "Grace Period" setting. (ie Vehicles must leave the parking site within X minutes of payment or be subject to additional fees. X will vary depending on site (Typically 15-30 minutes)

3. Local Agent – Comms Module Communications Link.

3.1 Communications between the Local Agent and the Comms Module shall be by IP messaging.

3.2 It is expected that the Local Agent will act as a client, while the Comms Module will act as a server.

3.3 One or two dedicated ports shall be used for communications.

3.3.1 Exact Port Number(s) to be determined later.

3.4 It is expected that if the communications port fails, the Local Agent shall automatically try to re-establish the communications link.

4. Local Agent – Comms Module Commands

4.1 The following commands may be sent from the Local Agent to the Comms Module.

4.1.1 Get Ticket Value (Command "G") – Comms Modules shall reply with the expected Parking Fee for the specified Barcode Ticket.

4.1.2 Authorize Payment (Command "A") – Comms Modules shall apply full payment to the specified Barcode Ticket and reply with a Receipt Number.

4.1.3 Make Reconciliation Report (Command "M") – Comms Modules shall generate a reconciliation Report for a specified range of Dates.

4.1.4 Idle (Command "$") – This message can be sent periodically by the Local Agent to inform the Comms module that the Local Agent is still running properly.

4.2 Standard Command Structure for communications shall be as follows

4.2.1 STX | CMD | DATA | CHECKSUM | ETX |

4.2.1a STX = Fixed byte = 0x02

4.2.1b ETX = Fixed byte = 0x03

4.2.1c CHECKSUM = 1 byte
= xor of CMD and all DATA bytes.
If value is less than 0x20 then add 0x20
Eg If xor of CMD and DATA results in 0x0A then CHECKSUM = 0x2A

4.3 Messages from Comms Module to Local Agent

4.3.1 The following messages / commands may be sent from the Local Agent to the Comms Module.

4.3.1.1 Synchronize Time (Command "T") – Comms Module may periodically send this message to the Local Agent so that both systems' Clocks can be synchronized.

4.3.2 Format of messages from Comms Module to Local Agent shall be as described in Section 4.2 above

5. Local Agent Command Details

5.1 Get Ticket Value (Command "G")

5.1.1 When the Parking system receives this command, it will return the Ticket #, Entry Time, parking Fee + other relevant value of the ticket

5.1.2 This command is only valid for unpaid / uncompleted Tickets.

5.1.3 DATA = Ticket Barcode.(20 ASCII characters)

5.1.4 Command example
STX | "G" | "10933291210461381762" | 0x4C | ETX

5.1.5 Parking system will reply with
STX | CMD | DATA | CHECKSUM | ETX |
Where :

5.1.5.1 CMD = Fixed byte = "G" or 0x47

5.1.5.2 DATA = ODATA + TICKET + ENTRY + EXIT + VALUE

5.1.5.2a ODATA (20 chars)
= Original barcode data that was received by the system

5.1.5.2b TICKET (8 Chars)
= Ticket number derived from ODATA

5.1.5.2c ENTRY (10 chars)
= Entry Date derived from ODATA in format: "YYMMDDHHNN"

5.1.5.2d EXIT (10 chars)
= Exit date (ie date of calculation) in format: "YYMMDDHHNN"

5.1.5.2e VALUE (6 chars)
= Calculated Parking Fee of ticket based on ENTRY and EXIT
= Value will be left padded with zeros
Eg If Value = 3000, then system will send "003000"

5.1.6 Response examples

5.1.6.1 STX | "G" | "10933291210461381762" | "01223344" | "1607011300" | "1607011435" | "002500" | 0x4B | ETX

5.1.6.1a Ticket Number = 01223344

5.1.6.1b Entry Time = 1300 (1pm) on 1st July 16

5.1.6.1c Fee Calculation Time = 1435 (2.35pm) on 1st July 16

5.1.6.1d Calculated Fee = 2500

5.1.6.2 STX | "G" | "10933291210461381762" | "ERORB001" | "0000000000" | "0000000000" | "000000" | 0x35 | ETX

5.1.6.2a Barcode sent is an Invalid Barcode (See below).

5.1.7 Notes

5.1.7.1 If there is an error calculating the Ticket Value, the TICKET data field will contain the letters "EROR" plus the error code. All other fields will be set to "0"

5.1.7.1.1 Error codes are as follows

"B001" = Invalid Barcode – Barcode is not a valid Maxpark parking barcode or is from a different site.

"C010 = Command Checksum Error – Checksum on command is incorrect.

"D002" = Mismatched Data – Data on barcode does not match data in system database

"D003" = Invalid Checksum – Checksum on barcode is incorrect.

"D008" = Invalid Ticket – Ticket # not found in system database

"D009" = Database Error – Unable to access system database.

"D012" = Invalid Message Format – Message is not in correct format

"P004" = Paid Ticket – Payment already received for this ticket, and is still within Exit Grace Period

"P005" = Used Ticket – Transaction already completed for this ticket (ie Vehicle has already left site)

"P015" = Paid Ticket but Ticket already expired (ie Past exit Grace Period).

"E999" = Unknown Error – Any other error condition that prevents the system from calculating the ticket value

5.1.7.1.2 EG If the barcode sent is not a valid Maxpark parking barcode then the TICKET field will contain "ERORB001" and all other fields will be set to "0"

5.1.7.2 It may take some time for the system to calculate the Parking Fee if the vehicle was parked for a few days, so it is advised that any timeout for this command should be > 5 seconds

5.1.7.3 It is possible that when making payment, the actual Parking Fee will be LOWER than the Fee calculated earlier (eg "Early Bird"). In this case, any additional payment is lost.

~~5.1.7.3 UID mentioned above is Unique ID and is equivalent to the Payment Reference specified in "Specification Document for Carpark Integration"~~

~~5.1.7.4 Customers will still have to present the ticket to a payment station (Autopay or manual payment station) for payment even if the parking fee is reduced by discounts to zero~~

~~5.1.7.5 It is possible that when making payment, the actual Parking Fee will be LOWER than the Fee calculated earlier (eg "Early Bird"). In this case, the Ticket will be considered free and any additional discounts are lost.~~

5.1.8 e-Tickets

5.1.8.1 In e-Ticket transactions, the customer will not have a physical ticket, but instead an 8-digit e-Ticket number will be sent directly to the payment App.

5.1.8.2 In this case DATA for the "G" command will be modified as follows:

5.1.8.2a DATA = e-Ticket Information.(20 ASCII characters)
= "ETICKET" (Fixed) + 8-Digit e-Ticket # + " " (5x Spaces)

5.1.8.3 Command example
STX | "G" | "ETICKET81234567 " | 0x2E | ETX

5.1.8.4 Response from the Comms Module will remain as described above.

5.2 Authorize Payment (Command “A”)

5.2.1 When the Parking system receives this command, it will apply full payment to the appropriate Ticket and return a Receipt / Payment number.

5.2.2 Currently, partial payments will not be allowed

5.2.3 This command is only valid for uncompleted Tickets.

5.2.4 DATA = Ticket Number (8 ASCII characters), Payment Authorization Code (12 Chars), Payment Amount (6 Chars)

5.2.5 Command Example
STX | “A” | “01123456” | “ABCDEFGH1234” | “001000” | 0x4A | ETX

5.2.6 Parking system will reply with
STX | CMD | DATA | CHECKSUM | ETX |
Where

5.2.6.1 CMD = Fixed byte = “A” or 0x41

5.2.6.2 DATA = TICKET + RECEIPT + VALUE + GST + PDATE + GRACE + STATUS

5.2.6.2a TICKET (8 Chars)
= Ticket number that was received by the system

5.2.6.2b RECEIPT (10 Chars)
= Receipt Number for the Payment.

5.2.6.2c VALUE (6 Chars)
= Parking Fee for the ticket in sen (ie Payment Value)

5.2.6.2d GST (4 Chars)
= GST value calculated by parking system (included in VALUE).

5.2.6.2e PDATE (12 Chars)
= Date / Time of payment in format: “YYYYMMDDHHNN”

5.2.6.2f GRACE (3 Chars)
= Grace Period for the Paid Ticket in minutes. Value will be left padded with 0’s
= If value is 999, then Grace period is until end of day (ie 2359 hours)

5.2.6.2g STATUS (4 chars)
= Status of Authorize Payment command.

5.2.6.2.1 Status update codes are as follows:

"S000"	= Payment applied OK
"C010"	= Command Checksum Error – Checksum on command is incorrect.
"D008"	= Invalid Ticket – Ticket # not found in system database
"D009"	= Database Error – Unable to access system database.
"D012"	= Invalid Message Format – Message is not in correct format
"P004"	= Paid Ticket – Payment already received for this ticket, and is still within Exit Grace Period.
"P005"	= Used Ticket – Transaction completed for this ticket (ie Vehicle has already left site)
"P006"	= Used UID – UID has already been registered by the system NB. UID is equivalent to Payment Authorization Code.
"P013"	= Underpayment – Payment Authorization value is less than Parking Fee
"P014"	= Used Authorization Code – Payment Authorization Code already used before.
"P015"	= Paid Ticket but Ticket already expired (ie Past exit Grace Period).
"E999"	= Unknown Error – Any other error condition that prevents the system from applying the discount.

5.2.7 Response Examples

5.2.7.1 STX | "A" | "01123456" | "RP12345678" | "001000" | "0057" | "201610011345" | "030" | "S000" | 0x18 | ETX

5.2.7.1a Payment successfully applied to Ticket Number 011232456.

5.2.7.1b Receipt Number = RP12345678

5.2.7.1c Fee paid = 1000

5.2.7.1d GST value = 57 (ie Actual Fee = 943 and GST = 57)

5.2.7.1e Ticket paid on 1st Oct 2016 at 13.45

5.2.7.1f Grace period is 30 miniutes

5.2.7.2 STX | "A" | "01112233" | "0000000000" | "000000" | "0000" | "000000000000" | "000" | "P004" | 0x14 | ETX

5.2.7.2a Payment not applied as ticket has already been paid (error code P004)

5.2.8 Notes

5.2.8.1 The Carpark system will keep track of all Payment Authorization Codes received and shall store them locally until manually purged.

5.2.8.2 Size of Payment Authorization Code is assumed to be 12 chars.

5.2.8.3 The Carpark system will allow payments to be greater than the current Parking Fee.

5.3 Make Reconciliation Report (Command "M")

5.3.1 When the Parking system receives this command, it will generate an "Online Payment Report" as an ASCII text file and store it in a specified folder in the local server

5.3.2 It is expected that the Local Agent will retrieve this file and send it to the cloud.

5.3.3 It is also expected that the Local Agent remove the file after retrieval to avoid filling up the folder.

5.3.4 DATA = Report Start Date (8 chars) in format "YYYYMMDD" +
Report End Date (8 chars) in format "YYYYMMDD"

5.3.5 Command Example
STX | "M" | "20160101" | "20160131" | 0x4E | ETX

5.3.6 Parking system will reply with
STX | CMD | DATA | CHECKSUM | ETX |
Where

5.3.6.1 CMD = Fixed byte = "M" or 0x4D

5.3.6.2 DATA = RSDATE + REDATE + STATUS + FILENAME

5.3.6.2a RSDATE (8 chars)
= Original Report Start Date that was received by the system ("YYYYMMDD").

5.3.6.2b REDATE (8 chars)
= Original Report End Date that was received by the system ("YYYYMMDD").

5.3.6.2c STATUS (4 chars)
= Status of Report Generation

5.3.6.2c.1 Status codes are as follows:
"S000" = Report Created OK
"C010" = Command Checksum Error – Checksum on command is incorrect.
"D011" = No data found for the requested date range
"D012" = Invalid Message Format – Message is not in correct format

5.3.6.2d FILENAME (12 chars)
= Name of the report file. In general filename will be in the following format:
"R" + YYYYMMDD + SEQ
Where SEQ = 3 digit sequence number from 000 to 999
Eg – "R20160131001"

5.3.7 Response Examples

5.3.7.1 STX | "M" | "20160701" | "20160702" | "S000" | "R20160703001" | 0x4F | ETX |

5.3.7.1a Report date range from 1st Jul 2016 to 2nd Jul 2016

5.3.7.1b Report generated successfully in file R20160703001.RPT

5.3.7.2 STX | "M" | "20160703" | "20160703" | "D011" | "000000000000" | 0x33 | ETX |

5.3.7.2a Report date range from 3rd Jul 2016 to 3rd Jul 2016

5.3.7.2b Report not generated as no data was found.

5.3.8 Report Format shall be as follows:

SITE CODE
SITE NAME
REPORT START DATE
REPORT END DATE
REPORT GENERATION DATE/TIME

PAYMENT DATE/TIME PA CODE RECEIPT # AMT PAID

5.3.9 Report shall be sorted by Payment Date/Time

5.3.10 Notes.

5.3.10.1 PA CODE above refers to the Payment Authorization Code received from the Online Payment Server

5.3.10.2 Report will be stored as an ASCII text file

5.3.10.3 Report Filename will have the .RPT extension. (Eg. R20160131001.RPT)

5.3.11 Sample Report as follows:

Site Code : SABCD
Site Name : Site ABCD
Report Start : 01-Jul-2016
Report End : 31-Jul-2016
Report Date : 01-Aug-2016 10:30

PAYMENT DATE/TIME	PA CODE	RECEIPT #	AMT PAID
01/07/2016 09:30	ABCDEFGH	P1R123456789	RM7.50
01/07/2016 10:44	12345678	P2R123123123	RM5.00
01/07/2016 11:09	12312312	R1R123456790	RM5.00

5.4 Idle Message (Command “$”)

5.4.1 This message is sent by the Local Agent to inform the Parking Server that it is still alive and working properly.

5.4.2 It is expected that the Local Agent will send this message once every 10-30 seconds if it is not busy.

5.4.3 The Parking Server will normally not reply to this message.

5.4.3.1 However, if the Date / Time included in this message varies greatly from the current Parking Server Date / Time (normally >= 5 minute difference), then the Parking Server may send a Synchronize Time Command (see section 6.1.1) to the Local Agent

5.4.4 DATA = Local Agent Date / Time (12 Chars) in format “YYYYMMDDHHNN”

5.4.5 Command Example
STX | “$” | “201601101735” | 0x21 | ETX

5.4.6 Special “Loopback” DATA.

5.4.6.1 If DATA content sent by Local Agent is “000000000000” then the Comms Module will reply with the same message. Ie STX | “$” | “000000000000” | 0x24 | ETX

6 Parking Server messages

6.1 Synchronize Time (Command “T”)

6.1.1 This is a Command from the Comms Module to the Local Agent

6.1.2 When the Local Agent receives this message, it should re-configure its local clock to match the Data setting of the command.

6.1.3 DATA = Parking System Clock (12 chars) in format “YYYYMMDDHHNN”

6.1.4 Command Example

STX | “T” | “201601271345” | 0x56 | ETX

6.1.5 When Local Agent receives the above Command, it should change its system clock to 27th Jan 2016 13.45.00.

Status Codes

“S000”	= Discount updated OK
“B001”	= Invalid Barcode – Barcode is not a valid Maxpark parking barcode or is from a different site.
“C010	= Command Checksum Error – Checksum on command is incorrect.
“D002”	= Mismatched Data – Data on barcode does not match data in system database
“D003”	= Invalid Checksum – Checksum on barcode is incorrect.
“D008”	= Invalid Ticket – Ticket # not found in system database
“D009”	= Database Error – Unable to access system database.
“D011”	= No data found for the requested date
“D012”	= Invalid Message Format – Message is not in correct format
“D016”	= Ticket Checksum Error – Error in printed ticket checksum (CALE system only).
“P004”	= Paid Ticket – Payment already received for this ticket.
“P005”	= Used Ticket – Transaction completed for this ticket (ie Vehicle has already left site)
“P006”	= Used UID – UID has already been registered by the system
“P007”	= Overpayment – Discount value is more than the value of the ticket.
“P013”	= Underpayment – Payment Authorization value is less than Parking Fee
“P014”	= Used Authorization Code – Payment Authorization Code already used before.
“P015”	= Paid Ticket but Ticket already expired (ie Past exit Grace Period).
“E999”	= Unknown Error – Any other error condition that prevents the system from performing the task.

B=Barcode
C=Command
D=Data
P=Procedure
S= Status
Last Number Used = 16