

タイムカードソフトの作成とその運用

—勤務時間の見える化と働き方改革への挑戦—

教頭 谷上正明

働き方改革の推進が学校現場においても喫緊の課題となっている。本校においてもこれまで在校時間記録の提出を義務づけてきたが、報告時間はあくまで自己申告によるものであり、客観的な記録とは言えない状況にあった。平成 30 年 1 月に大学よりタイムカード導入の方針が告げられ、より効率的な運用を考える中でマイクロソフト「エクセル」の VBA を活用しようと考えた。今回は VBA のコード（プログラム）を含めた仕様の解説、運用状況、問題点等を報告したい。

<キーワード>タイムカード ICT エクセル VBA 働き方改革

1. はじめに

公立学校の教師の勤務時間の上限に関するガイドライン（平成 31 年 1 月 25 日：文部科学省）が示され、その実施に当たっては「タイムカードによる記録，電子計算機の使用時間の記録等の客観的な方法その他の適切な方法による勤務時間の把握が事業者の義務として明確化されたことを踏まえ，在校時間は，ICT の活用やタイムカード等により客観的に計測し，校外の時間についても，本人の報告等を踏まえてできる限り客観的な方法により計測すること。」と留意事項が指示されている。

本校においては、平成 30 年 1 月より PC ソフトを利用した出退勤時刻の記録がなされている。様々な問題点はあるものの、比較的容易に使用でき、記録データについても有効に活用されている状況である。この報告では、具体的な設置状況や使用しているエクセル VBA のコードの紹介、退勤時間の変化などについて報告し、今後に向けた問題提起としたい。

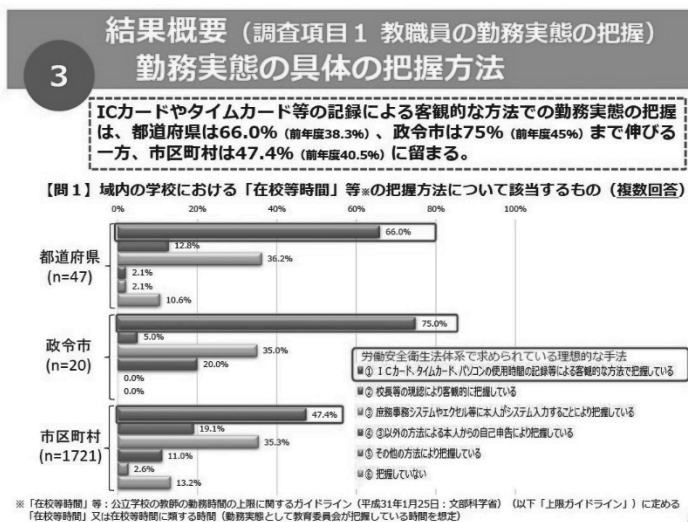


図 1 文部科学省資料

2. これまでの流れ

(1) 平成 29 年 8 月以前

各教員が自己申告により、出勤時間、退勤時間、勤務時間外業務の内容と時間を報告していた。

(2) 平成 29 年 9 月～12 月

大学事務局からの指示により、勤務時間外については「業務従事」と、自己研鑽等による「施設利用申請」とに分けて報告していた。報告についてはすべて自己申告であった。

(3) 平成 30 年 1 月～3 月

次年度 4 月より「機械式タイムカード」導入の方針が示され、附属高校では「PC によるタイムカー

ド」による記録を試行的に開始した。月末にはこれまで同様「施設利用申請および出退勤時刻記録」の提出がされたが、出退勤時刻については、あらかじめPCに入力された電子データを取り込んで提出するものに改められた。

(4) 平成30年4月～

PCによるタイムカード記録について大学事務局より許可を頂き、正式に運用を開始した。なお、平成30年度より教職調整額に該当する手当が廃止され、4月より超過勤務時間を申請し、残業代として支給されることとなった。

3. システム構成

システムの運用に当たり、打刻用、データ保存用、データ保守管理用、個人端末の4種類のPCが稼働している。まず、これらについて説明しておきたい。

(1) 出退勤時刻の打刻用パソコン（共用PC）

職員室西側入り口に、各職員が出退勤時に時刻を記録するためのデスクトップパソコンを1台設置する。なお、校内ネットワークに接続されており、他のアクセス許可をされたパソコンからデータアクセスが可能にしておく。また、ネットワーク不調時およびサーバーのメンテナンス時でも記録できるよう、記録データは一端このPC内に保存される。

(2) 校内データサーバー（Windows Sever）

バックアップデータをサーバー内の共同作業用フォルダに保存し、各教員用PCから出退勤データが自由に活用できるようにする。

(3) 出退勤時刻の保守・管理用パソコン（教頭PC）

打刻忘れへの対応や記録データのバックアップ作成用に、教頭パソコン内にメンテナンス用のエクセルファイルを入れておく。

(4) 各教員の個人端末

校内データサーバーに用意されている各教員用の「超過勤務申出書兼時間外学校施設利用申請書」に、各自でサーバーに記録されている出退勤時刻データを取り込み、電子ファイルを提出する。

4. PCによるタイムカードソフトについて

先述のようにタイムカード導入に際し、大学からは機械式タイムカード機が提案された。これは紙製カードに出退勤時刻を印刷していくもので、以前から各事業所等で利用されてきたものである。しかし、終業時刻になると何人分かのカードをまとめてパンチするといった不正が横行する危険が予想された。また、カードに直接パンチすることで電子データは存在せずデータの流用性が低く、月末には各教員が超過勤務申出書や時間外学校施設利用申請書の作成が必要なことを考えると、その際手書きで時刻を書き写す必要があり、大変作業効率が悪い。そこで、不正に打刻されることが少なく、より正確な時間を記録し、かつ記録したデータを有効に活用できる方法はないかと考えた。

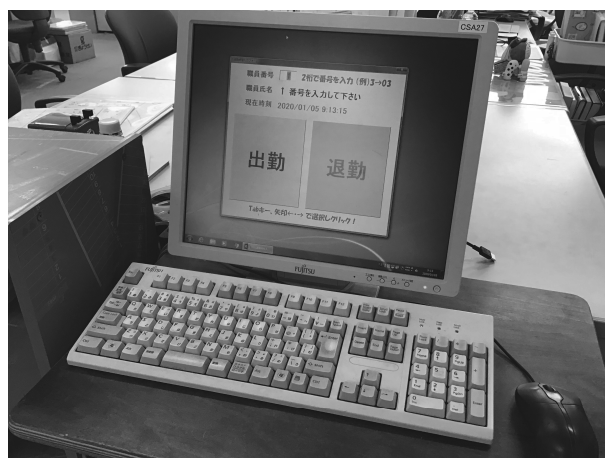


図2 職員室での打刻用PC設置の様子

は、不正に打刻されることが少なく、より正確な時間を記録し、かつ記録したデータを有効に活用できる方法はないかと考えた。

市販されているソフトの使用も考えたが、急なことで予算確保がされていなかったこと、超過勤務申請書などにデータを流用することが難しいことも考え合わせ、自作することとした。なお、自作に当たり、①誰でも使えるソフトであること、②比較的容易なプログラムであること、③既存のシステムを活用できること、を考え、マイクロソフトのエクセルVBAを活用することとした。以下、実際に作成した3種類のエクセルファイルについて紹介する。

(1) タイムカード(打刻)用ファイル

①使用方法、設置の実例

各教員は出勤時と退勤時に年度ごとに割り振られる「職員番号」を入力し、「出勤」または「退勤」をクリックする。各時刻は打刻用PC内の記録用シートに入力され、その都度自動的に上書き保存される。出張等で早い時間に学校を離れる場合は、退勤と同じ扱いとしている。また、土日など部活動で出校する場合は、職員室が解錠されておらず打刻できない場合もあるので、一律打刻しないこととしている。

②VBAコードおよび解説

書面スペースの都合もあり、ここでは「画面に職員氏名を表示させる(図4)」、「出勤時刻記録(図5、8)」についてのみ紹介する。



図3 打刻画面

<職員氏名表示>

```

1 Private Sub TextBox1_BeforeUpdate(ByVal Cancel As MSForms.ReturnBoolean)

    Dim X As Long
    nendo = Worksheets("年度").Range("A1")
    Ndata = "Ndata" & nendo

5 On Error GoTo 0

    X = TextBox1.Value
    If X = 99 Then 'コード99でエクセルを終了する
        Application.Quit
    ElseIf X < 1 Or 36 < X Then
10 With TextBox1
        .SetFocus
        .Value = 1
        .SelStart = 0
        .SelLength = Len(TextBox1)
15 End With
        MsgBox ("職員番号は1から35までの数字を入力してください。")
        Label3.Caption = "↑番号を入力して下さい"
        Cancel = True
    Else
20 Label3.Caption = Application.WorksheetFunction.VLookup(X, Worksheets(Ndata).Range("name_data"), 2, False)
    End If
End Sub

```

図4 職員氏名表示用のコード

・7行目：VBAは誤作動防止のため通常終了しないように設定されている。コード「99」を入力するとエクセルを強制的に終了するようにした。

- ・ 9 行目：職員番号を 1 から 3 5 に入力制限しエラー回避する。
- ・ 20 行目：Application.WorksheetFunction.Vlookup は通常のエクセル関数 Vlookup を VBA 上で有効にし、職員番号から職員氏名を画面上に表示している。

< 出勤時刻記録 >

```

1 Private Sub CommandButton1_Click()

    Dim 出勤時刻 As String
    Dim a As Integer ' 入力行番号
    Dim b As Long ' 本日の日付データ (シリアル値)
5   Dim X As Long ' 入力データ (職員番号)
    Dim Wb_T As Workbook
    Dim Ws_T As Worksheet
    Dim MyColor As Long
    Dim rc As VbMsgBoxResult

10  MyColor = CommandButton1.BackColor
    CommandButton1.BackColor = RGB(100, 100, 255)

    On Error Resume Next
    Application.ScreenUpdating = False
    nendo = Worksheets("年度").Range("A1")
15  Ddata = "Ddata" & nendo
    kiroku_n = "kiroku" & nendo & ".xlsx"
    出勤時刻 = Format$(Time(), "hh:nn")
    X = TextBox1.Value
    b = Worksheets(Ddata).Range("D1").Value2 ' Value2 で日付けをシリアル値に変換
20  a = Application.WorksheetFunction.VLookup(b, Worksheets(Ddata).Range("A1:C400"), 3)
    Mypath = ThisWorkbook.Path
    Filepath = "¥●残業・施設利用申請¥" & nendo & "¥tool¥"
    ' Netpath = "¥¥○○○○¥附属高校¥○○¥○○" & Filepath
    Netpath = "C:¥Users¥Public¥Documents¥"

```

図 5 出勤時刻記録用のコード(前半)

- ・ 10、11 行目：出勤をクリックした際にボタンの色を変化させる。
- ・ 13 行目：処理が終了するまで画面表示を一端止める。
- ・ 14～20 行目：処理する年度、日付、打刻時間、書込用シート名 (図 7)、何行目に書き込むのかなどをメモリー上に格納する。
- ・ 21～24 行目：作業ファイルのフォルダ名を指定している。なお、コードの一部はセキュリティー上の理由により○○とした。また、日付と行番号は行番号参照シート (図 6) を参照しており、B列の日付に対してA列のシリアル値、C列の行番号に対応させている。

	A	B	C	D	E	F	G	H
1	43556	2019/4/1	1	2020/1/2				
2	43557	2019/4/2	2					
3	43558	2019/4/3	3	A列にシリアル値を入力。				
4	43559	2019/4/4	4	検索はこの列を見て、行番号を使用している。				
5	43560	2019/4/5	5					
6	43561	2019/4/6	6					
7	43562	2019/4/7	7					
8	43563	2019/4/8	8					
9	43564	2019/4/9	9					
10	43565	2019/4/10	10					
11	43566	2019/4/11	11					
12	43567	2019/4/12	12					
13	43568	2019/4/13	13					
14	43569	2019/4/14	14					
15	43570	2019/4/15	15					
16	43571	2019/4/16	16					

図 6 行番号参照シート

	U	V	W	X	Y	Z	AA	AB	AC	AD
243	8:24	20:30			8:02	12:01			7:13	11:28
244	8:26	17:35			8:10	17:31	8:02	16:30	7:14	15:33
245										
246										
247			7:58	15:03	8:29	20:54	7:52	17:10	7:22	17:36
248	8:27	17:49	8:04	17:03	8:27	21:26	8:07	18:32	7:21	19:46
249	8:27	17:48	7:41	17:04	8:25	20:37	7:39	17:58	7:14	20:12
250	8:25	19:09	7:43	17:16	8:28	21:10	7:15	18:58	7:13	19:44
251	8:26	17:58	7:28	17:04	8:28	19:13	8:07	18:58	7:13	18:36
252										
253										
254	8:25	19:16	7:44	18:18	8:22	20:27	7:58	18:50	7:08	19:22
255	8:24	18:00	7:49	17:02	8:23	12:04	7:39	17:56	7:06	18:02
256	8:55	20:12	7:56	17:04	8:28	20:23	8:09	18:17	7:11	18:47
257	8:24	18:28	8:07	18:17	8:26	20:38	7:06	18:16	7:11	19:17
258	8:24	17:52	7:01	14:00	8:24	13:40	8:01	18:24	7:08	19:36

図 7 時刻の記録用紙シート

```

25 Workbooks.Open Netpath & kiroku_n
   Set Wb_T = Workbooks(kiroku_n)
   Set Ws_T = Wb_T.Worksheets("TimeRec")
   If Ws_T.Cells(a, 2 * X - 1).Value = "" Then
     Ws_T.Cells(a, 2 * X - 1).Value = 出勤時刻
30 Wb_T.Close Savechanges:=True
   Application.ScreenUpdating = True
   MsgBox "おはようございます。" & vbCrLf & "出勤時刻を記録しました。"
   Else
     rc = MsgBox("すでに【出勤時刻】が入力されています!" & vbCrLf & "データを上書き更新しますか?", vbYesNo +
       vbDefaultButton2 + vbExclamation, "上書き確認!!")
35 If rc = 6 Then
     Ws_T.Cells(a, 2 * X - 1).Value = 出勤時刻
     Wb_T.Close Savechanges:=True
     Application.ScreenUpdating = True
     MsgBox "おはようございます。" & vbCrLf & "出勤時刻を記録しました。"
40 Else
     Wb_T.Close Savechanges:=False
     Application.ScreenUpdating = True
   End If
End If
45 With TextBox1
  .SetFocus
  .Value = 1
  .SelStart = 0
  .SelLength = Len(TextBox1)
50 End With
Label3.Caption = "↑ 番号を入力して下さい"
CommandButton1.BackColor = MyColor
End Sub

```

図 8 出勤時刻記録用のコード(後半)

- ・ 25～27 行目：書込用シート（図 7）を開く。
- ・ 28～39 行目：打刻の有無を調べ、すでに入力されていれば上書きするかどうか確認し、入力が必要であれば新しい時刻を書き込みし、「おはようございます」とメッセージを表示する。
- ・ 41 行目以降：開いていたシートを保存しながら閉じ、止めていた画面表示を元に戻して、再度初期画面を表示させている。

③工夫点、課題・問題点

年度が変わるごとに職員名簿や日付データ更新に手間をかける必要がないよう、別シートで容易に追加できるようにした。また、年度制度は4月をスタート月とするため、それをシステムに認識させることは工夫が必要であった。

(2) 打刻管理用ファイル

①使用方法、設置の実例

打刻忘れの再入力、年休や割振等に伴う打刻時間の修正、データのバックアップなど日々の作業は意外に多い。それらを短時間で確実に実施するためにメンテナンス用のファイルを作成した。

「入力済データ一覧」(図 9)、「修正ツール」(図 10)、「データ表示」(図 11)の3種類のシートを用意してある。各シートの使用法は次の通りである。

- ・ 入力済データ一覧：特定日の打刻状況を一覧で表示し、未入力者には「打刻修正依頼書」を提出してもらいデータ訂正をする。

職員番号	氏名	出退時刻	職員番号	氏名	出退時刻	職員番号	氏名	出退時刻
1			13		17:12	25		17:39
2		17:27	14		11:42	26		17:20
3			15		15:04	27		12:56
4		16:22	16		18:44	28		18:04
5			17		16:02	29		18:45
6		17:05	18		15:08	30		
7		20:32	19		17:09	31		17:17
8		18:15	20		17:16	32		17:28
9		18:19	21			33		
10		16:15	22			34		18:55
11			23			35		18:29
12			24			36		

図 9 入力済データ一覧シート

- ・修正ツール: 打刻忘れ等の職員の修正日を表示させる。バックアップ作成用のボタンも配置してある。
- ・データ表示: 修正ツールで指示された日や期間を表示し、修正後、書込、保存をする。

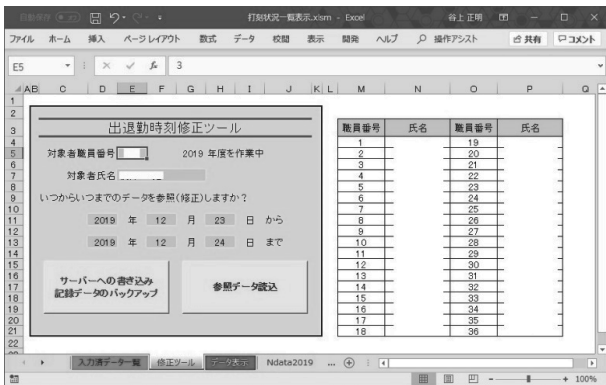


図 10 修正ツールシート

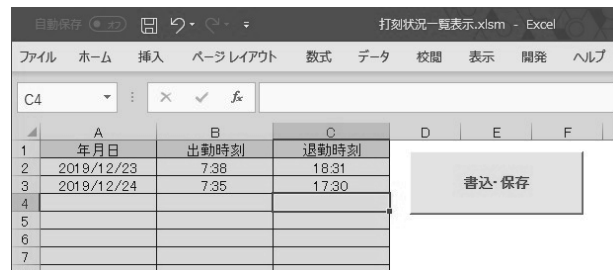


図 11 データ表示シート

②VBAコードおよび解説

```

1  MyDateS = DateSerial(Ys, Ms, Ds)
   MyDateE = DateSerial(Ye, Md, De)
   MyDateB = MyDateE - MyDateS + 1

   Dim Wb_M As Workbook, Wb_S As Workbook
5  Dim Ws_M As Worksheet, Ws_S As Worksheet, Ws_C As Worksheet

   MyPath = ThisWorkbook.Path
   MyName = ThisWorkbook.Name
   NetPath = "¥¥CSA27¥Users¥Public¥Documents¥"
10  NetPath_s = "¥¥○○○○¥附属高校¥○○¥○○¥●残業・施設利用申請¥" & nendo & "¥tool¥"
   Ddata = "Ddata" & nendo
   kiroku_n = "kiroku" & nendo & ".xlsx"
   kiroku_nb = "kiroku" & nendo & "(backup).xlsx"

   Set Wb_M = Workbooks(MyName)
   Set Ws_M = Wb_M.Worksheets("データ表示")
15  Set Ws_C = Wb_M.Worksheets(Ddata)

   MyCol = Application.WorksheetFunction.Match(MyDateS, Ws_C.Range("A:A"), 0)

   Ws_M.Range("A2:C400").ClearContents

   On Error Resume Next
   Application.ScreenUpdating = False

20  Workbooks.Open NetPath & kiroku_n
   Set Wb_S = Workbooks(kiroku_n)
   Set Ws_S = Wb_S.Worksheets("TimeRec")

   For i = 1 To MyDateB

   Ws_M.Cells(i + 1, 1).Value = Ws_C.Cells(MyCol + i - 1, 2).Value
25  Ws_M.Cells(i + 1, 2).Value = Ws_S.Cells(MyCol + i - 1, 2 * a - 1).Value
   Ws_M.Cells(i + 1, 3).Value = Ws_S.Cells(MyCol + i - 1, 2 * a).Value

   Next i

   Wb_S.Close Savechanges:=False
   Wb_M.Worksheets("データ表示").Activate
30  Wb_M.Worksheets("データ表示").Range("A1").Select
   Wb_M.Worksheets("データ表示").Range("B2").Select

   Application.ScreenUpdating = True

```

図 12 参照データ読込コード

全コードは長いので、主要コードのみ紹介する。

- ・ 1～3行目：参照期間の行数を計算する。
- ・ 6～15行目：参照ファイルとシートをメモリー上に格納する。
- ・ 16行目：Application.WorksheetFunction.Match は通常のエクセル関数 Match を VBA 上で有効にし、参照データの該当行を入手する。
- ・ 20～22行目：記録用紙シート（図7）を開いておく。
- ・ 24～26行目：参照データを画面上（図11）に書き込んでいく。

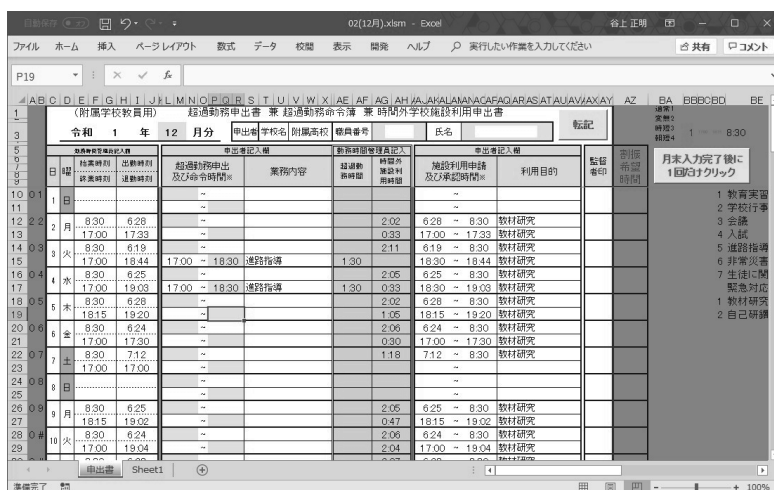
③工夫点、課題・問題点

現在では打刻忘れもほとんどなく順調に運用できているが、運用開始当初は打刻忘れが少なくなかった。できるだけ効率よく作業する必要があり何度かの改良を経て現在の形になった。ただ、データサーバーの更新時期にはコード変更の必要があり、残念ながら誰にでも使えるファイルにはなっていない。今後の後任者による作業を考えるとより簡略化されたシステムの必要性を強く感じている。

(3) 超勤申請書（勤務時間報告書）へのデータ取込ファイル

①使用方法、設置の実例

各教員が主に月末報告時に使用するファイルである。右上氏名の右側に「転記」ボタンを配置し、クリックすることで該当月の出勤時刻と退勤時刻を簡単に取り込むことができる。また、欄外には入力完了後に提出するための「送信」ボタンが配置されている。超過勤務時刻と施設利用時刻を入力後、クリックすることで提出が完了する。



②VBAコードおよび解説

2種類のコード、「転記」（図14）と「送信」（図15）について紹介したい。

図13 超過勤務申請書兼施設利用申請書

<転記>

- ・ 9～14行目：ファイル名やシート名、格納パス名などをメモリー上に格納する。
- ・ 17および22行目：必要なデータが入力されたシートを開いておく。
- ・ 25～36行目：セルにデータが入力済みの時はそのまま飛ばし、空白セルのみ時刻データを順番に記入していく。

<送信>

- ・ 8行目：提出用のファイル名をシート上の入力項目から作成する。
- ・ 10行目：シートをPDF形式で印刷し、これをファイル出力として所定のフォルダに保存する。なお、このコードは校内の他の文書にも使用されており、ネットワーク上での書類提出に活用され大変便利なコードでもある。
- ・ 11行目：提出が完了したメッセージを表示する。

```

1 Sub ボタン1_Click()

    a = Range("AG3").Value '対象者
    y = Range("G3").Value + 2018 '対象年
    m = Range("K3").Value '対象月
5 Dim Wb_M As Workbook, Wb_S As Workbook
    Dim Ws_M As Worksheet, Ws_S As Worksheet, Ws_C As Worksheet
    Dim MyCol As Integer '入力対象行番号
    Dim MyDate As Long '対象月1日のシリアル値
    Mypath = ThisWorkbook.Path
10 MyName = ThisWorkbook.Name
    MyDate = DateSerial(y, m, 1)
    Netpath = "¥¥○○○○¥附属高校¥○○¥●残業・施設利用申請¥2019¥tool"
    Set Wb_M = Workbooks(MyName)
    Set Ws_M = Wb_M.Worksheets("申出書")

15 On Error Resume Next

    Application.ScreenUpdating = False
    Workbooks.Open Netpath & "¥Ddata2019.xlsx"
    Set Wb_S = Workbooks("Ddata2019.xlsx")
    Set Ws_S = Wb_S.Worksheets("Ddata2019")
20 MyCol = Application.WorksheetFunction.Match(MyDate, Ws_S.Range("A:A"), 0)
    Wb_S.Close Savechanges:=False
    Workbooks.Open Netpath & "¥kiroku2019.xlsx"
    Set Wb_R = Workbooks("kiroku2019.xlsx")
    Set Ws_R = Wb_R.Worksheets("TimeRec")

25 For i = 1 To 31
    If Ws_M.Cells(2 * i + 8, 3).Value <> "" Then
        If Ws_M.Cells(2 * i + 8, 8).Value = "" Then
            Ws_M.Cells(2 * i + 8, 8).Value = Ws_R.Cells(MyCol + i - 1, 2 * a - 1).Value
        End If
30 End If
    If Ws_M.Cells(2 * i + 8, 3).Value <> "" Then
        If Ws_M.Cells(2 * i + 9, 8).Value = "" Then
            Ws_M.Cells(2 * i + 9, 8).Value = Ws_R.Cells(MyCol + i - 1, 2 * a).Value
        End If
35 End If
Next i

    Wb_R.Close Savechanges:=False
    Application.ScreenUpdating = True

End Sub

```

図 14 出退勤時刻の転記用コード

```

1 Sub 仮提出()

    Dim fileName As String
    Folname = "¥¥○○○○¥附属高校¥共同作業用フォルダ (誰でも変更・削除できます) ¥【仮保存文書】 ¥"
    fname = "超勤申請" & Range("G3").Value & Range("K3").Value & " "
5 bname = Range("AG3").Value
    cname = Range("AN3").Value
    sdate = Date & " " & Time

    fileName = Folname & fname & bname & cname & ".pdf"
    Range("I28").Value = fileName
10 ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, fileName:=fileName

    MsgBox (sdate & " " & Range("K3").Value & "月分の超勤申出書を送信 (提出) しました。")

End Sub

```

図 15 入力後のデータ送信用コード

③工夫点、課題・問題点

各教員が申請書を作成するために残業しては本末転倒である。できる限り入力を簡単にし、省力化できるようなシート作成を考えた。出退勤時刻をワンタッチで取り込み、超過勤務時間や施設利用申

請を簡単に入力できるよう何度も改良を加えた。今回は紹介できなかったが、シート上の関数の扱いについても工夫した。ただ、入力を簡単にしたいがためにコードが複雑化してしまえば、システムがブラックボックス化してしまう。エクセルの知識がある方であれば理解できるよう、できる限りシンプルなコード作成に留意したつもりである。

5. 勤務時間の変化

最後に、これまでの取り組みを通して勤務時間がどのように変化してきたかを見ておきたい。本来であれば在校時間について検証すべきと思うが、残念ながら在校時間の形ではデータが残っておらず退勤時間の比較による検証を試みた。(退勤時刻と出勤時刻の差をとれば可能だが膨大なデータ量を手入力することになり断念した。)

次表に挙げる数値は、平成 29 年 4 月から令和元年 12 月までの全教員の平均退勤時刻である。なお、退勤時刻の平均算出に当たり、時短勤務教員や年休、出張等に伴う明らかな終業時刻前の退勤記録は除外をした。

年度 \ 月	4月	5月	6月	7月	8月	9月	10月	11月	12月	1月	2月	3月
H29	19:06	18:59	19:02	18:45	17:29	18:55	18:46	18:48	18:25	18:57	18:31	18:35
H30	19:09	19:09	19:00	18:49	17:38	19:06	19:01	18:51	18:53	18:56	18:40	18:39
H31(R01)	19:14	19:13	19:09	19:02	18:02	19:12	19:07	19:02	19:03			

表 1 退勤時刻変化

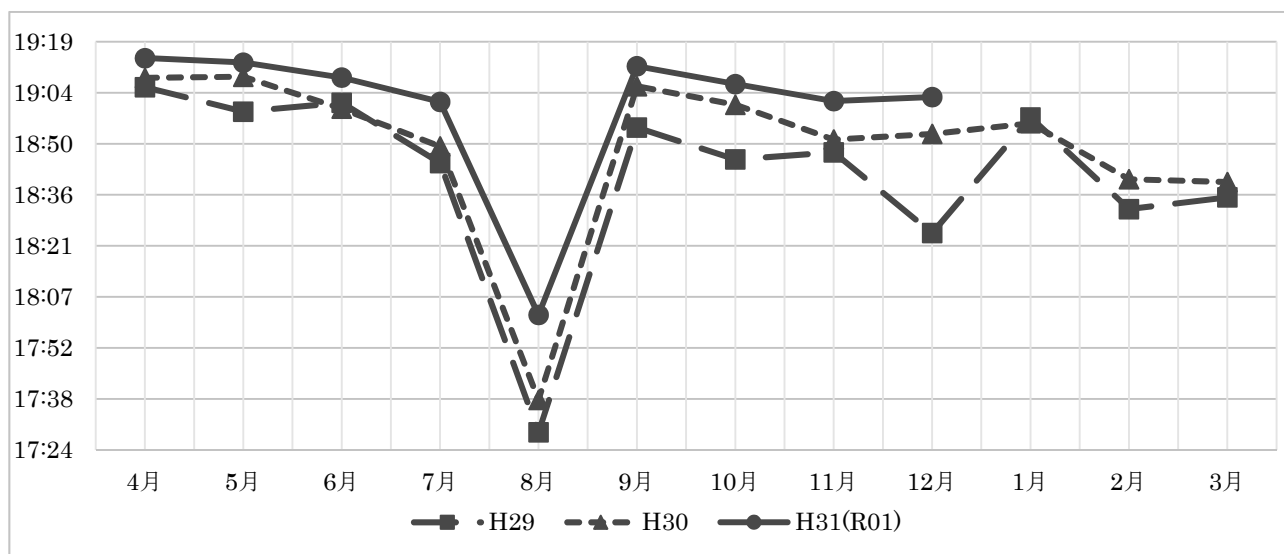


図 16 退勤時刻変化グラフ

PCソフトを利用した出退勤時刻の記録により在校時間の見える化を進めることができた。しかし、実際の結果は徐々に「退勤時刻が遅くなっている」という残念な結果になってしまった。在校時間を各教員はもちろん、管理職が把握することは大変重要なことである。タイムカードを導入すればよい、ICTを活用すればよい、ということではなく、いかに改善につなげていくかが重要であることを改めて認識するよい機会となった。

6. 今後の課題とまとめ

学校現場における働き方改革が声高に叫ばれ、改善に向けた取り組みが随所で行われている。愛知県

においても、県教委を中心に各校でタイムカード導入に向けた計画が動き出している。これまでの学校現場は、出勤簿に押印し管理職が目視で勤務状況を把握するといった状況にあった。勤務状況を把握し、実態が見える化する上ではタイムカードは重要なツールとなるだろう。しかし、導入すれば改善されるというものではなく、多忙化の原因はどこにあるのか、その軽減のためにどのように取り組んでいくのが大切である。今回、PCソフトを利用した出退勤時刻の記録によりその問題点をはっきりさせることができた。ソフト自体もまだまだ問題点も多く、改善の必要があるが、それ以上に学校全体で取り組んでいく重要性を再認識した。地域の研究実践校として、これらの成果を活用しよりよい学校教育実現に向け努力していきたいと思う。

7. おわりに

タイムカードソフトの作成と運用に当たり、愛知教育大学教授飯島康之前校長、附属高校北河洋一校長には大学事務局との調整等、ご多忙の中お時間を割いて頂きました。また、本校情報科の先生方にもパソコンの設置、ネットワークやサーバーの設定など大変お世話になりました。深く感謝の意を表します。

引用文献・参考文献

- 1) 文部科学省(2019) 令和元年度教育委員会における学校の働き方改革のための取組状況調査